Cost optimized multipath transmission of bursty video traffic in 5G multi-access network architecture

Markus Amend

Supervisor: Dr Veselin Rakocevic

Department of Engineering
City, University of London

This dissertation is submitted for the degree of
Doctor of Philosophy

October 2023

During this work, I had to deal with life, and life had to deal with me. I am very grateful for that experience!

I would like to dedicate this thesis to my loving wife and children…

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 60.000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 70 figures.

Markus Amend
October 2023

Acknowledgements

When I made the decision some years ago to pursue a doctorate, two little boys and my wife already accompanied me through life, I was in the beginning of my professional career in industry and we planned to build a house. With this multitude of activities, which often required higher priority, the time available for this work melted away.

It is thanks to many people and the right amount of luck in life that I can look back and see that everything turned out wonderfully.

I would like to thank Professor Dr.-Ing. Joachim Habermann and Professor Dr.-Ing. Karl-Friedrich Klein, who motivated me to take the next step after my studies. In addition to their enormous efforts to provide me with a professional set of tools, they never forgot the human aspects and the importance of perspective. The support of Mr. Habermann, who was still available as a contact person during this work, cannot be taken for granted either.

Another big thank you I want to dedicate to Dr.-Ing. Gerhard Kadel. As my first boss in industry, he gave me the support, flexibility and trust without which it would have been impossible to start the doctorate in parallel with my professional work.

Who deserves a predominant part of my thanks is Dr Veselin Rakocevic. Without hesitation, he agreed to be available as a doctoral supervisor, opening the door for this work. During the many years, he was always available when it required a helping hand, the right advice or a push. I am sure that the relationship that has developed will last beyond this work.

For those for whom it was most difficult I would like to say a special thank you. My wife Isabell and my children Benedikt and Martin gave me the time I needed to complete this work. It was never easy, but with the right management from my wife and the support of my parents Gabriele and Klaus and my parents-in-law Angelika and Kurt, it worked.

The long educational path from my apprenticeship as an electrician to today would have to let me name many people here who deserve to get a thank you for helping me achieve the impossible. All who have accompanied me on this path and read these lines are thanked.

Abstract

In modern multiservice networks, with terminals equipped with multiple network interfaces, there is a clear trend to move from the dominating single path transport towards multipath. There are obvious benefits of the multipath service delivery – these include better resilience and improved throughput - and the standardization of multipath transport protocols MP-TCP, Multi-path Datagram Congestion Control Protocol (MP-DCCP), Multi-path Quick UDP Internet Connections (MP-QUIC) and their usage in the 3GPP rel. 16 (and beyond) 5G ATSSS (Access Traffic Splitting, Steering and Switching) multipath framework pave the way for broad implementation. While the field of traffic distribution algorithms for multipath transport is subject of extensive research, this work addresses the challenge of cost-based optimisation of scheduling in the multipath 3GPP ATSSS context. The work demonstrates that there is a major conflict for the Video-on-Demand (VoD) traffic between the achievable Quality of Experience (QoE) and the consumed multipath resources when a simple path prioritization algorithm – e.g. the *Cheapest-Path-First (CPF)* – is used to direct traffic. Using real network and testbed measurements and months of trials with mobile phone users, this work shows that for Video-on-Demand (VoD) in multipath up to 90% of the expensive path resources are consumed while QoE does not take any advantage from this, primarily because of the natural burstiness of the VoD traffic. The work then proposes a novel service transparent and lightweight *Cost-Optimized-Multipath (COM)* traffic scheduling algorithm. Using extensive measurement of YouTube video streams and an MP-TCP implementation of the COM scheduler, this work demonstrates that – by finding the right balance between the QoE and the incurred costs - the new scheduler can provide better QoE compared to the single path transport, while eliminating the spurious resource consumption on the expensive path.

Table of contents

List of figures

List of tables

List of listings

List of algorithms

Nomenclature

Symbols

$\Delta\theta$	Delta between two cost factors θ
γ	Collective term for any QoE gain
$\gamma_{initial}$	Gain between two $QoE_{initial}$ measurements
$\gamma_{resolution}$	Gain between two $QoE_{resolution}$ measurements
γ_{stall}	Gain between two QoE_{stall} measurements
$\mathfrak{F}$	Generic expression for functions
θ	Cost factor of a data transmission over a composite multipath *mc*
B_i	Throughput or Bandwidth of an individual path within a multipath connection *mc* [bps]
B_p	Throughput or Bandwidth of a transmission path [bps]
B_{cheap}	Throughput or Bandwidth of the cost efficient path within a composite multipath connection *mc* [bps]
B_{demand}	Throughput or bandwidth requirements of a service, e.g. VoD [bps]
B_{mc}	Throughput or Bandwidth of a composite multipath connection *mc* [bps]
B_{SCP}	Throughput or Bandwidth corresponding to B_{cheap} but indicating its single usage outside *mc* [bps]
B_{Ui}	Utilized Throughput or Bandwidth of an available B_i [bps]

B_{Umc}	Utilized Throughput or Bandwidth of an available $B_m c$ [bps]
C_i	Path cost of an individual path within a multipath connection *mc*
C_{cheap}	Path cost of the cost efficient path within a composite multipath connection *mc*
C_{mc}	Cost of data transmission over a composite multipath *mc*
ch	Chunk of video frames
C	Path cost of a path p_i
J_i	Jitter of an individual path within a multipath connection *mc* [s]
J_{mc}	Jitter of a composite multipath connection *mc* [s]
J	Jitter [s]
L_i	Latency of an individual path within a multipath connection *mc* [s]
L_{mc}	Latency of a composite multipath connection *mc* [s]
L	Latency [s]
mc	Composite multipath connection
M	Video segmentation range for creating video chunks *ch*
N	Number of individual paths within *mc*
p_i	Individual transmission path within *mc*
$QoE_{initial}$	QoE during the initial request of VoD
$QoE_{resolution}$	Resolution dependent QoE of VoD
QoE_{stall}	Interruption dependent QoE of VoD
QoE_{VoD}	QoE of VoD
R_i	Loss rate of an individual path within a multipath connection *mc* [bps]
R_{mc}	Loss rate of a composite multipath connection *mc* [bps]

$RecvBuffer_{VoD}$ Client buffer for holding incoming video data that is received faster than the playback rate s of the VoD stream [pkts] or [B]

R Loss rate [bps]

r Video frame(s) resolution [px]

s Video playback rate [Hz]

$t_{144...1920px}$ Time in which a video is played back within $t_{playback}$ with a certain resolution [s]

T_{block} Time of the last block event where COM prevents access to the expensive path [s]

t_{ch_end} Time of the last data of a video chunk at a certain measuring point during transmission [s]

t_{ch_start} Time of the first data of a video chunk at a certain measuring point during transmission [s]

t_{ch} Duration in time of a video chunk defined by a start t_{ch_start} and end time t_{ch_end} [s]

T_{Delay} Time COM scheduler prevents access to an expensive path in a composite multipath connection mc [s]

$T_{GAPrecv}$ Time interval between transmissions of successive video chunks at the receiver [s]

$T_{GAPsend}$ Time interval between transmissions of successive video chunks at the sender [s]

$T_{GAPthresh}$ A threshold time configured in COM scheduler of a measured T_{GAP} after which the expensive path in a composite multipath connection mc is blocked by T_{Delay} [s]

T_{GAP} Time interval between transmissions of successive video chunks at a certain measuring point [s]

$t_{initial}$ Time duration until the start of a video after the playback request [s]

$T_{lastdata}$ Time of the last COM data scheduling event [s]

$t_{measurement}$	Measurement time of a video transmission, including initial request, stall events and playback [s]
$T_{notraffic}$	A period of time within $T_{GAPthresh}$ in which any transmitted data is resetting the gap calculation. [s]
$t_{playback}$	Time during which a video is played back, excluding stall and first request event [s]
t_{stall}	Amount of time a requested video stops during playback [s]
$T_{StartDelay}$	Like T_{Delay} but used by COM at the beginning of a connection without a T_{GAP} calculation required [s]
$T_{Volume,max}$	A period of time within $T_{GAPthresh}$ in which data up to V_{max} can be transmitted without resetting the gap calculation. [s]
tp_{VoD}	Traffic pattern of a VoD service
V_{ch}	Data volume of video chunk *ch* [B]
V_{max}	Maximum number of allowed packets/volume during $T_{Volume,max}$ [pkts] or [B]
V_{sum}	Volume counter during $T_{Volume,max}$ [pkts] or [B]

Acronyms / Abbreviations

3GPP	3rd Generation Partnership Project
4G	4th generation mobile networks
5G	5th generation mobile networks
ABR	Adaptive Bit Rate
ATSSS	Access Traffic Steering Switching Splitting
AVC	Advanced Video Coding

BBF	Broadband Forum
BBR	Bottleneck Bandwidth and Round-trip
BDP	Bandwidth Delay Product
CC	Congestion Control
COM	Cost Optimized Multi-path
CWND	Congestion Window
CPE	Customer Premises Equipment
CPF	Cheapest-path-first
CMT-SCTP	Concurrent Multipath Transfer SCTP
DASH	Dynamic Adaptive Streaming over HTTP
DCCP	Datagram Congestion Control Protocol
DSL	Digital Subscriber Line
GRE	Generic Routing Encapsulation Protocol
HA	Hybrid Access
HLS	HTTP Live Streaming
HTTP	Hypertext Transfer Protocol
HoL	Head-of-Line
IP	Internet Protocol
LTE	Long Term Evolution
MBB	Mobile Broadband
MDC	Multiple Description Coding
MNO	Mobile Network Operator
MOS	Mean Opinion Score
MP-DCCP	Multi-path Datagram Congestion Control Protocol

MP-QUIC	Multi-path Quick UDP Internet Connections
MPTCP	Multipath TCP
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
OS	operating system
OTT	Over the top
PDU	Protocol data unit
QoE	Quality of Experience
QUIC	Quick UDP Internet Connections
RG	Residential Gateway
RTMP	Real Time Messaging Protocol
RTO	Retransmission Time-Out
RTT	Round-Trip-Time
SCP	Single Cheap Path
SCTP	Stream Control Transmission Protocol
SVC	Scalable Video Coding
SRTT	smoothed Round-Trip-Time
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UE	User Equipment
UPF	User Plane Function
VoD	Video-on-Demand
Wi-Fi	Wireless Local Area Network
WWC	Wireless Wireline Convergence

Chapter 1

Introduction

While the Internet connectivity today is mainly provided over a single access network, the last decade has seen deployments which have demonstrated that multi-connectivity - simultaneous usage of more than one access technology - is able to offer better Quality of Experience (QoE) for a range of applications [1, 2, 3]. Multi-connectivity enables a cost-efficient utilisation of network resources, while at the same time improving connection resilience and overcoming shortcomings of the single access technology. Aggregated capacities of multiple access paths are made available, providing a platform for the users of multi-connectivity not to be affected by the interrupting handovers between access technologies or limiting throughput capabilities of the single access.

This chapter brings in context existing multi-connectivity frameworks and deployments and their cost efficient operation in terms of traffic distribution. An observation in such networks, however, motivates in section 1.1 a new field of research because of the inadequacy of current mechanisms to ensure cost reduced multi-path transmission. This leads to the formulation of the research questions and objectives in section 1.2 and section 1.3, which also serve as the basis for the described structure in section 1.6 of this document. That the author is capable of carrying out such work is shown in section 1.7, which is also reflected in a first summary of the contributions and results in section 1.4 and section 1.5.

1.1 Motivation and Overview

The cost efficiency of multi-connectivity solutions is one of the dominant requirements from the network operators' point of view. As a rule, the transmission costs per bit in 4G or 5G mobile networks are higher than for fixed-network connections, resulting in one cheap and one expensive transmission path when a device can choose between the two types of access. In [4], an economic benefit for customers and operators of mobile networks is identified if traffic can be shifted to Wireless Local Area Network (Wi-Fi). Similar is confirmed in [5]. For 5G networks [6] and 5G based Fixed Wireless Access [7] it is shown that the cost per bit goes down due to better spectral efficiency. Due to the higher operating costs of mobile networks and their limited capacity, especially in rural areas, the balance still swings in favor of fixed access in most scenarios. One of the first large commercial deployments of multi-connectivity, known as Hybrid Access (HA) [8] (based on Generic Routing Encapsulation Protocol (GRE) protocol [9], and introduced in Germany in 2014), made use of the Cheapest-path-first (CPF) scheduling principle to maximise utilisation of the fixed access pipe for traffic delivery, switching to the more expensive cellular access only when the fixed access pipe became saturated.

In the networks of today, the limitations of the GRE approach in supporting multiple radio links mean that more mature and scalable support for multi-connectivity, provided by Multipath TCP (MPTCP) [10], is required. The inherent path measurement of MPTCP offers efficient service to multipath traffic management in 5th generation mobile networks (5G) networks, leading to standardisation developments within the 3rd Generation Partnership Project (3GPP) Access Traffic Steering Switching Splitting (ATSSS), specified first in the 3GPP Rel. 16 [11]. The 5G ATSSS is a multi-connectivity framework for mobile user equipment such as smartphones, but it has also been adopted in the meantime for HA [12], substituting GRE. Similar to HA, 5G ATSSS multi-connectivity is terminated in the network operators core at a Proxy. The Proxy ensures transparent conversion between the multi-connectivity and traditional single-path transport towards the final destination without requiring service adaptation. MPTCP's obvious limitation to Transmission Control Protocol (TCP) services has recently been complemented by the standardisation work on MP-DCCP [13] and MP-QUIC [14] to support the non-TCP services. This is currently being discussed to be integrated into the enhanced ATSSS [15].

The basic components of a multipath system for simultaneous path usage are depicted in Figure 1.1. The transmission over at least two paths between a Sender and a Receiver requires a sender-side traffic distribution logic - a Scheduler [16]. To avoid overloading individual path capacities, timely information about the available path capacity - provided by a path estimation entity - must be known to the scheduler. This is typically done based on measurement (e.g. congestion control in MPTCP/MP-DCCP/MP-QUIC), or in static setups using e.g. the Digital Subscriber Line (DSL) synchronization rate. Typically the scheduler is agnostic to the other components of a multipath system as long as the required input parameters can be provided for the selected scheduling logic.

At the receiver side, an almost mandatory feature is the re-assembly of the transmitted data unless the carried service has no demand on in-order delivery. Simultaneous transfer over different paths typically leads to scrambling of the original packet order, requiring the use of a re-ordering module, which takes care of that based on the sequencing information. In MPTCP, this follows the TCP inherited principle of strict in-order delivery using re-transmission. For the GRE-based HA without re-transmission, re-ordering is time- or buffer-clocked and therefore not strictly delivering packets in order. A multipath network protocol takes care of the transmission, but can also provide sequencing, re-ordering and path estimation, like MPTCP does.

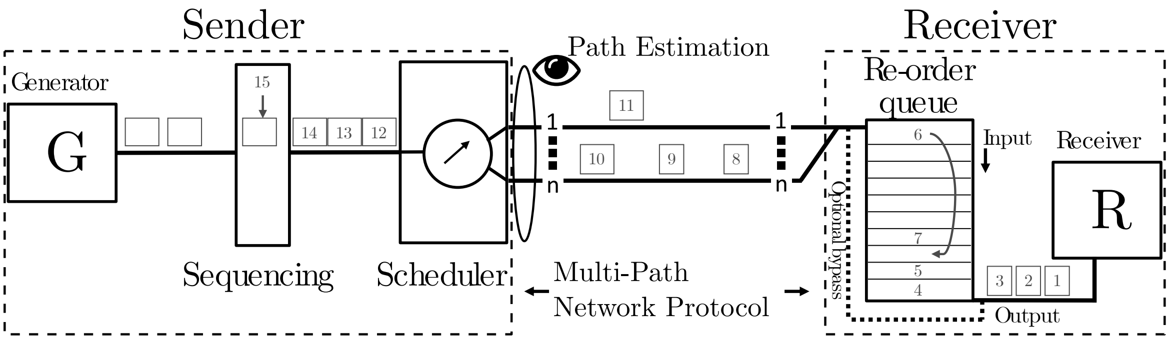


Fig. 1.1 Components of a multipath system with path estimation and scheduler, multipath transport protocol and sequencing for re-ordering

Clearly, the transmission cost in a multipath system is dominated by the decision of the scheduler how often traffic is sent over an expensive path (e.g., cellular) instead of using a cheaper path (e.g., using Wi-Fi or fixed access). For example, the Cheapest-path-first scheduling as used in the Hybrid Access scenarios should optimize the transmission cost as it first saturates the cheap fixed access path before overflowing into the expensive cellular access path.

However, what sounds like a simple solution to keep cost under control, fails in reality in the presence of significant amount of Video-on-Demand traffic. In Figure 1.2 a measurement of a commercially deployed GRE based HA in Germany, operated by a Tier 1 ISP and using CPF scheduling, is shown. Two measurements were performed with a typical HA connection with 16 Mbit/s DSL and up to 50 Mbit/s LTE, transmitting a Video-on-Demand service (purple line) on the one hand and a linear TV service (magenta line) on the other. VoD traffic significantly consumes cellular network resource due to its bursty nature which generates short but significant throughput demands. Contrary to this, a "flat" demand like linear TV, file downloads etc. is typically kept in the fixed access.

Considering the dominance of VoD in today's Internet (estimated by Cisco in 2021 [17] at 80% of all traffic, and confirmed by more recent studies such as [18] and [19]), the challenge

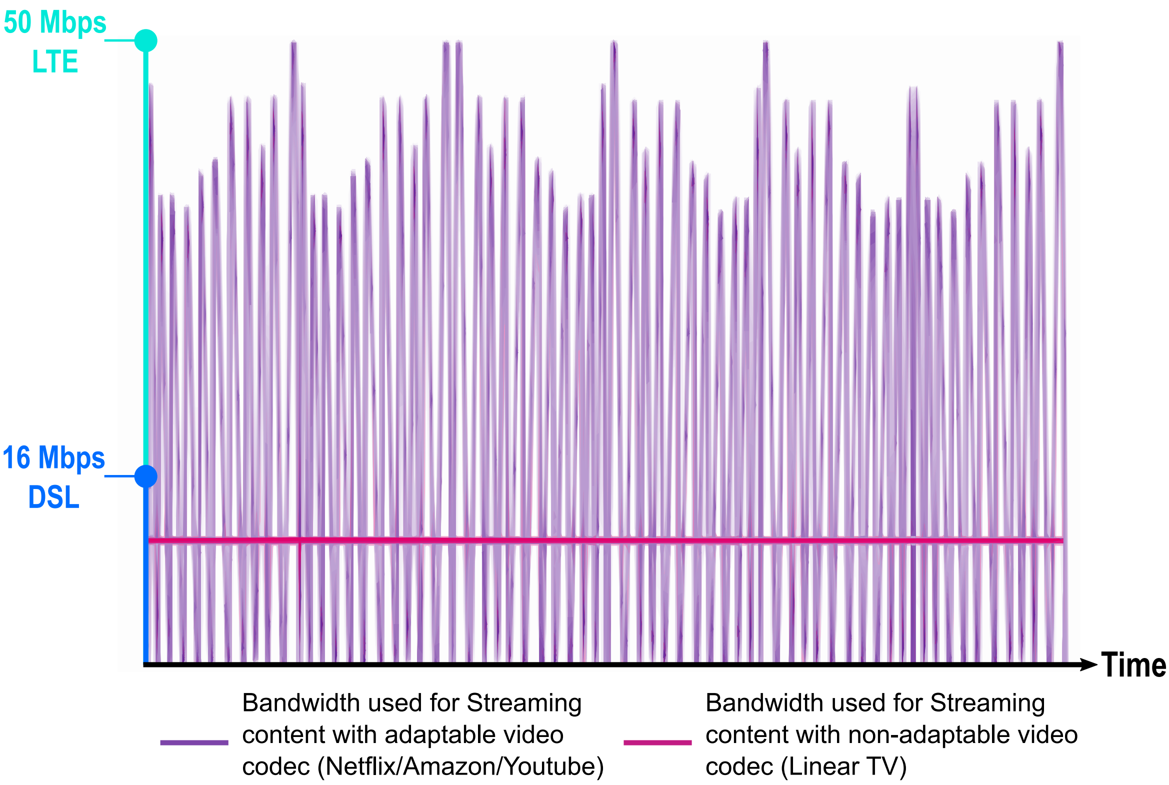


Fig. 1.2 Cost contradicting Video-on-Demand traffic in Hybrid Access with preferred DSL line over LTE access

of adopting multi-connectivity standards and solutions to the features of VoD traffic stands out. In this dissertation a new scheduling solution is offered, designed especially for VoD traffic. Results of extensive tests demonstrate that this new Cost Optimized Multi-path scheduling can make a significant difference in comparison to CPF. As an extension of the Cheapest-Path-First principle, Cost Optimized Multi-path (COM) identifies VoD traffic and suppresses aggregation of a higher cost access path as long as QoE is not compromised. In typical Hybrid Access scenarios, this most often eliminates or at least reduces by a quarter the consumption of cellular resources, which is otherwise 20% to 90% for 1080p videos as the comparison between the COM scheduler and the CPF scheduler in chapter 5 shows.

1.2 Research questions

As described earlier, multi-connectivity networks are on the rise due to their seamless integration into operator networks without changing the services carried. The commercial use of Hybrid Access and the upcoming 5G ATSSS and Broadband Forum (BBF) WWC suggest further proliferation. On the other hand, customer of such operators consume more and more bandwidth hungry services which give rise to the desire on both the customer and the operator side, to deliver such services as cost efficient as possible to keep the cost for all parties low. An important factor that goes into the calculation of cost-effectiveness is the QoE, which in multi-connectivity networks is to be increased beyond the performance of a single access path.

In HA and ATSSS, the multipath scheduler (Figure 1.1) takes care of this cost efficient operation where an identified cheaper access path is first filled before an expensive access path is used to carry the data which exceeds the capacity of the first path. This simple logic is plausible and helps on paper to keep traffic away from expensive transmissions. However, first commercial implementations showed a surprisingly high use of the expensive access channel and first analyses identified VoD, the most consumed service on the Internet, as the culprit. Even more surprising was the finding in these initial analyses that the expensive path was aggregated even if these services were previously transmitted without problems via the single cheaper path when multi-connectivity was not yet offered.

Therefore, the spurious demand on high capacities provided by multi-connectivity systems is identified as a key issue if the requirement on such a system is to reduce the costly resource footprint to a minimum while keeping the QoE of customers and services higher than compared to single-path environments.

So far, there is evidence that the simple access cost-based prioritisation in the multipath schedulers used or specified in HA and ATSSS is not efficient in all cases and causes costs that counteract the actual goal of minimising costs. On the other hand, it is known about the increasing HA customer numbers [20], which indicate better QoE than with DSL alone. Therefore, the following scientific and technical questions arise.

Q1 **Are the observed spurious requests for expensive path resources in HA and ATSSS multi-connectivity networks with prioritization of cheap path resources provable and traceable to specific patterns?**

Approach: Detailed study of the scenario in a multi-connectivity test environment using a cost-prioritizing multipath scheduler, including identification of services that generate spurious expensive capacity demand.

Background: From previous observation it is known that website requests or video-on-demand services like YouTube and Netflix trigger spurious demands and overload costly resources in existing service transparent multi-path frameworks of HA or using MPTCP as defined in ATSSS.

Q2 **In the landscape of existing multi-path solutions, are there answers to the question of how to curb the occurrence of spurious expensive requests?**

Approach: State-of-the art research considering especially available network scheduling approaches and elaboration of the problematic key characteristics of identified services.

Background: When it can be expected, that the operator of a multi-connectivity system is not the owner of the service requested by a customer, the scheduling in multi-connectivity systems seems to be the only control mechanism available to reach resource usage optimization. Protocol data unit (PDU) scheduling is a long lasting topic in networks evolution, in particular if bottlenecks arises or more than one resource is available, so different strategies exist for a lot of challenges. How much they are related to the existing issue is subject matter of this investigation. Special attention should be spent on PDU schedulers which profit from a detailed path knowledge as it is the case within MPTCP used for HA and ATSSS.

Q3 **What is a viable technical solution that does not require interaction with the service, that promises an efficient way to avoid unwanted expensive requests without decreasing QoE?**

Approach: Based on boundary conditions create design principles and develop thereof strategies to overcome or mitigate unnecessary costly capacity overflows.

Background: Based on services and/or traffic patterns causing spurious capacity demands, combined with the knowledge of existing scheduling approaches, one or more strategies shall be developed to get rid of unnecessary costly resource usage.

Q4 **What are the barriers to integrating such a technical solution?**

Approach: Implementation of one or several such strategies for usage in a HA and ATSSS relevant multi-path solution.

Background: Depending on the number of strategies found and maybe based on a developed criteria catalogue, one or more of them will be implemented. Preferably, this is done in a system with an open network stack that allows for easy modification and re-use.

Q5 **Is the implemented solution superior to the current approach of dumb cost prioritization of access paths?**

Approach: Define evaluation criteria, testbed integration and proof of concept possibly extended by a field trial.

Background: The evaluation finally decides about a successful approach or not. Various types of measurements should underpin the effectiveness of the solution. An ultimate approach is to test under real network conditions with real customers.

1.3 Objectives of the PhD research

The following objectives, which are pursued in this dissertation, can be derived from the research questions posed in section 1.2.

- Investigate the observed problem of increased costs in multi-connectivity scenarios that do not appear to lead to better QoE and identify the causes why existing CPF scheduler fails.
- Exploration of possible existing solutions and verification of their applicability in the field of operator multi-connectivity solutions such as Hybrid Access and ATSSS.
- Creation of design criteria for a solution to overcome the inefficiency of CPF, development of a corresponding scheduling algorithm and selection of a network protocol for its implementation.
- Define the success criteria to demonstrate the benefits of the new solution and evaluate them.

1.4 Summary of contributions

The dissertation is motivated by a real world problem in operator provided multi-access networks which aim to achieve higher throughput for better customer QoE but find themselves exposed to significant costs. Remedy is finally provided by a new solution developed, which mitigates or eliminates such cost of multi-access usage which does not contribute to better QoE.

This work shows that the typical scheduling mechanisms used in multi-access networks, which purport to distribute traffic in a cost-efficient manner, fail in this sense for the majority of Internet traffic. In particular, traffic generated by VoD services is identified as a troublemaker, as this traffic generates bursts. These bursts significantly distort the balance between the costs of multiple access transmission to be incurred and the achievable QoE gain.

Existing multipath network protocols or frameworks for multi-access delivery does not consider this issue so far but point to the multipath scheduler as responsible entity which has the most cost impact by the distribution of traffic across the multiple accesses.

An analysis of schedulers used in academia or that aim to reduce HoL blocking or optimise cost, video QoE or both reveal some interesting approaches that could be useful for tackling the problem found. Clustering applied to the analysed schedulers, which distinguishes between a dependency on the transmitted service, leaves none of the existing solutions and ideas in the quiver. Specified multi-access networks such as ATSSS and Hybrid Access are service-transparent and exclude any interaction from the outset.

Taking these limitations into account, a new approach is therefore developed. The analysis of the problem provides an obvious idea which, after some optimisation and discussion steps, leads to a very practical and handy new multipath scheduler that is solely dependent on the shape of the scheduled traffic pattern. Essentially, it is the gap between the packets that determines whether the scheduler distributes the traffic to the more expensive access path.

With an implementation in MPTCP, a clear definition of costs and QoE, the claimed possible solution space and the developed test environments to generate meaningful results for ATSSS and Hybrid Access, the right evaluation framework is created.

In various measurement campaigns, it is demonstrated that a universal parameter set for COM exists that offers the right balance in almost all practical scenarios and only generates additional costs through multi-path transport if QoE benefits from it. This is achieved by significantly better utilisation of the more cost-effective access compared to the status quo without COM.

1.5 Summary of results

This work shows a significant conflict between cost prioritised multi-path scheduling and services that generate a traffic pattern of bursts. In particular, video streaming (VoD) and website downloads fall into this category. Due to the proliferation of multi-path networks such as HA and ATSSS and the majority of traffic generated by the identified problematic services, a solution that allows for more efficient cost adjustment is required.

This seems to be possible since initial measurements revealed that most often the single cheap path in a HA or ATSSS environment is sufficient to provide the service with a good QoE. An analysis of existing multi-path solutions and scheduling approaches has shown that a lot of research is being done in this area, but for the specific scenarios of ATSSS and HA, which are service-transparent, no solution is offered and is not helpful. On the other hand deep dives into one of the used multi-path protocols MPTCP for HA and ATSSS, the VoD transmission principle and the understanding of VoD QoE paved the way for a new scheduling idea: COM, a traffic pattern based extension of the cost prioritizing scheduling approach used so far.

Focused measurements with an implementation of the cost prioritisation scheduler with and without COM showed the possibility to eliminate up to 90 % of spurious VoD demand with COM without reducing QoE. The definition of the KPIs cost and QoE as the main objective, followed by the development of COM from an idea through implementation and testing allowed to determine the crucial parameters and dependencies of COM, which can be summarized to be the distance between consecutive bursts, the time how long the expensive path is blocked and the bandwidth of the cheap path.

Finally, the comparison of COM in a trial with nomadic customers of a Tier 1 operator implementing the ATSSS principle showed a cost reduction of one third when COM is used for an uncontrolled traffic mix with no change in QoE.

1.6 Structure of the Thesis

The general structure of the thesis follows the raised questions Q1-Q5 from the research questions of this thesis defined in section 1.2 and the derived objectives in section 1.3 and is divided into the following chapters:

1 Introduction:
First indications are given that multi-connectivity frameworks like Hybrid Access and ATSSS have conflicts with at least VoD traffic if cost reduction is the scheduling goal. Furthermore, this chapter serves to formulate the objective of this thesis, to structure the document, to summarise the contributions and results and to state the author's merits in the context of the research.

2 Fundamentals:
Analysis of existing multi-path solutions and schedulers and their relationship to cost prioritisation and Video-on-Demand. Their function within HA and ATSSS is also assessed and how Congestion Control is used to estimate path properties for efficient scheduling decisions. Finally, the technical implementation, operation and determination of QoE of VoD are examined.

3 System model, Problem Statement and Solution:
Provides the multi-path system model and identifies the basic conflict between VoD transmission and cost-based multi-path scheduling using a self-developed MPTCP scheduler with the simple cost prioritisation logic used in HA and specified for ATSSS. This is further used to formulate the algorithm design goals, presents the new COM algorithm, and discusses its integration in the current 3GPP ATSSS framework on the basis of a MPTCP implementation of COM.

4 Methodology and testbed:
Development of a methodology and testbed that enables the evaluation of COM with respect to the problematic cost prioritisation logic and the major troublemaker VoD. Different testbed variants from very simple controlled environments to those similar to the ATSSS and HA frameworks focus at different stages of development on verification and final proof with specific services, in particular VoD, or uncontrolled traffic mixes.

5 Results and Analysis:
Outlines the benefit of the new approach and demonstrate its usability in terms of cost and QoE in the field with real Internet services. Also possible interference with handling of services without VoD transmission characteristic is subject of investigation.

6 Conclusion:
Based on the findings of the measurement and trial campaign the usage of the new algorithm

as extension to the Cheapest-path-first principle is recommended for network operators of Hybrid Access and ATSSS where transmission cost and QoE are equally important.
Additional ideas are presented to further validate and improve the concept of COM within and beyond MPTCP.

1.7 Author's publications

In the course of the author's work in the field of multipath and ATSSS, several publications in this area were produced with his participation.

The author of this work is the author of **5** and co-author of **7** peer-reviewed conference or journal papers, inventor of **7** published patent families related to multipath scheduling and **37** patent families in the field of multipath communication, editor of **5** and co-author of **2** standardization contributions and creator of **1** open source project.

1.7.1 Conference Papers

Papers related to the PhD research objectives

- M. Amend, V. Rakocevic, and J. Habermann, "Cost optimized multipath scheduling in 5G for Video-on-Demand traffic", in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2021 [21]
- M. Amend and V. Rakocevic, "Cost-efficient multipath scheduling of video-on-demand traffic for the 5G ATSSS splitting function", *Computer Networks*, vol. 242, 2024, ISSN: 1389-1286 [22]

General multipath scheduler related papers

- M. Pieska, A. Rabitsch, A. Brunstrom, A. Kassler, and M. Amend, "Adaptive Cheapest Path First Scheduling in a Transport-Layer Multi-Path Tunnel Context", in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW '21, Virtual Event, USA: Association for Computing Machinery, 2021 [23]
- M. Pieska, A. Rabitsch, A. Brunstrom, A. Kassler, M. Amend, and E. Bogenfeld, "Low-delay cost-aware multipath scheduling over dynamic links for access traffic steering, switching, and splitting", *Computer Networks*, vol. 241, 2024, ISSN: 1389-1286 [24]

General multipath related papers

- C. Lange, C. Behrens, E. Weis, J. Kraus, S. Krauss, M. Grigat, H. Droste, T. Rosowski, T. Monath, C.-A. Bunge, E. Bogenfeld, M. Amend, N. Bayer, M. Dueser, F.-J. Westphal, and A. Gladisch, “Bridging the Last Mile”, in *Broadband Coverage in Germany; 10. ITG-Symposium*, Apr. 2016 [25]

- N. Bayer, A. T. Girmazion, M. Amend, K. Haensge, R. Szczepanski, and M. D. Hailemichael, “Bundling of DSL resources in home environments”, in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016 [26]

- M. Amend, V. Rakocevic, A. P. Matz, and E. Bogenfeld, “RobE: Robust Connection Establishment for Multipath TCP”, in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW ’18, Montreal, QC, Canada: Association for Computing Machinery, 2018 [27]

- M. Amend, E. Bogenfeld, M. Cvjetkovic, V. Rakocevic, M. Pieska, A. Kassler, and A. Brunstrom, “A Framework for Multiaccess Support for Unreliable Internet Traffic using Multipath DCCP”, in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, Oct. 2019 [28]

- N. R. Moreno, M. Amend, A. Brunstrom, A. Kassler, and V. Rakocevic, “CCID5: An Implementation of the BBR Congestion Control Algorithm for DCCP and Its Impact over Multi-Path Scenarios”, in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW ’21, Virtual Event, USA: Association for Computing Machinery, 2021 [29]

- R. Alfredsson, A. Kassler, J. Vestin, M. Pieskä, and M. Amend, “Accelerating a Transport Layer based 5G Multi-Access Proxy on SmartNIC”, in *Würzburg Workshop on Next-Generation Communication Networks (WueWoWas’22) :*, Würzburg University, 2022 [30]

- M. Amend, N. R. Moreno, M. Pieska, A. Kassler, A. Brunstrom, V. Rakocevic, and S. Johnson, “In-network Support for Packet Reordering for Multiaccess Transport Layer Tunneling”, in *2022 IEEE 11th IFIP International Conference on Performance Evaluation and Modeling in Wireless and Wired Networks (PEMWN)*, Nov. 2022 [31]

- F. Brisch, A. Kassler, J. Vestin, M. Pieskä, and M. Amend, “Accelerating Transport Layer Multipath Packet Scheduling for 5G-ATSSS”, in *KuVS Fachgespräch - Würzburg*

Workshop on Modeling, Analysis and Simulation of Next-Generation Communication Networks 2023 (WueWoWAS'23), 2023 [32]

1.7.2 Patent submissions

Multipath scheduler patents (EU)

- M. Amend and E. Bogenfeld, "A Communication System for Transmitting a Transmission Control Protocol Segment Over a Communication Network Using a Multipath Transmission Control Protocol, Corresponding Method and Computer Program", European pat. 3579500B1, Nov. 17, 2021 [33]
- M. Amend and E. Bogenfeld, "Techniques for Scheduling Multipath Data Traffic", European pat. 3544332B1, May 26, 2021 [34]
- M. Amend and E. Bogenfeld, "Techniques for Detecting Bursty Traffic Pattern Detection and Scheduling Multipath Data Traffic", European pat. 3796604B1, Jun. 14, 2023 [35]
- M. Amend and E. Bogenfeld, "A method and communication device for transmitting multiple data streams of different communication services over a multipath transmission system", European pat. 4042653A1, Aug. 17, 2022 [36]
- M. Amend, E. Bogenfeld, A. Brunstrom, M. Pieska, and A. Kassler, "System and method for multipath transmission", European pat. 4080836B1, Jun. 21, 2023 [37]
- M. Amend, E. Bogenfeld, A. Brunstrom, M. Pieska, and A. Kassler, "System and method for multipath transmission with efficient adjustable reliability", European pat. 4075742A1, Oct. 19, 2022 [38]
- M. Amend, "Techniques for an efficient use of redundant multipath scheduling", European pat. 4044526A1, Aug. 17, 2022 [39]

All patents of the author in the field of multipath are listed in Appendix B.

1.7.3 IETF Standardization

- M. Amend, E. Bogenfeld, A. Brunstrom, A. Kassler, and V. Rakocevic, "A multipath framework for UDP traffic over heterogeneous access networks", Internet Engineering Task Force, Internet-Draft draft-amend-tsvwg-multipath-framework-mpdccp-01, Jul. 2019 [40]

- M. Amend, A. Brunstrom, A. Kassler, and V. Rakocevic, “Lossless and overhead free DCCP - UDP header conversion (U-DCCP)”, Internet Engineering Task Force, Internet-Draft draft-amend-tsvwg-dccp-udp-header-conversion-01, Jul. 2019 [41]
- M. Boucadair, O. Bonaventure, M. Piraux, Q. D. Coninck, S. Dawkins, M. Kühlewind, M. Amend, A. Kassler, Q. An, N. Keukeleire, and S. Seo, “3GPP Access Traffic Steering Switching and Splitting (ATSSS) - Overview for IETF Participants”, Internet Engineering Task Force, Internet-Draft draft-bonaventure-quic-atsss-overview-00, May 2020 [42]
- O. Bonaventure, M. Piraux, Q. D. Coninck, M. Baerts, C. Paasch, and M. Amend, “Multipath schedulers”, Internet Engineering Task Force, Internet-Draft draft-bonaventure-iccrg-schedulers-02, Oct. 2021 [16]
- M. Amend and D. V. Hugo, “Multipath sequence maintenance”, Internet Engineering Task Force, Internet-Draft draft-amend-iccrg-multipath-reordering-03, Oct. 2021 [43]
- M. Amend and J. Kang, “Multipath TCP Extension for Robust Session Establishment”, Internet Engineering Task Force, Internet-Draft draft-amend-tcpm-mptcp-robe-02, Mar. 2022 [44]
- M. Amend, A. Brunstrom, A. Kassler, V. Rakocevic, and S. Johnson, “DCCP Extensions for Multipath Operation with Multiple Addresses”, Internet Engineering Task Force, Internet-Draft draft-ietf-tsvwg-multipath-dccp-13, Jan. 2024 [13]

1.7.4 Open Source

- MP-DCCP is a protocol for providing multipath transport for latency sensitive services and/or services with no or less demand on reliable delivery and optional adjustable re-ordering. While DCCP applications are natively supported, another approach is to provide multi-path transport for UDP or IP traffic and becomes therefore appealing for Hybrid Access and 3GPP ATSSS like scenarios. As part of the protocol development, an open source Linux reference implementation is available at https://multipath-dccp.org.

Chapter 2

Fundamentals

This chapter provides a basic understanding and related research studies of multipath transport in computer networks and the typical VoD-playout technique used on the Internet. In section 2.1 the genesis of multipath transport is reappraised and multipath scheduling approaches are discussed, clustered and assessed with regard to the objectives of this work. The multipath transportation relationships and requirements associated with the deployment scenario that is the focus of this work, 3GPP ATSSS, are discussed in section 2.2. For the transport of traffic in today's network topologies, most often Congestion Control (CC) is used to avoid fair usage of scarce transmission resources. The effect of CC on the today's dominating multipath network protocols is investigated in section 2.3. With VoD as the most prominent originator of traffic the principles of service provisioning and resulting traffic patterns are analyzed in section 2.4. In section 2.5 the performance indicators of VoD are elaborated along with a standardized method to determine QoE.

2.1 Multi-path concepts

The literature research shows a significant body of work in multipath. While most of the research and standardisation efforts are focused on the multipath transport layer protocols, significant work exists in the area of multipath scheduler optimization for a wide range of use cases. The development of MPTCP during the last decade has motivated most of these works, as MPTCP's congestion control mechanisms offer path characteristic measurements. Hence, multipath transport over heterogeneous and volatile networks can be managed and optimized by feeding the path characteristics into the multipath scheduler logic. At the same time, the video streaming use case spawned multipath scheduling considerations, either in conjunction with MPTCP but also in conjunction with protocols in other network layers or directly integrated into the video application.

This section aims to answer three central questions: First, are there existing multipath concepts that discuss or even solve the problem of unwanted demand on expensive access paths described in the motivation of this work (chapter 1)? Second, which existing concept is best suited to develop, implement and evaluate a solution to demonstrate its usefulness for ATSSS and Hybrid Access? Third, in the event that the first question does not provide a solution: What multipath scheduling strategies are known and what are their dependencies?

Developing multipath traffic delivery solutions for modern networks has a long history, dating back almost three decades. Table 2.1 gives a glimpse into the development of those solutions across the TCP/IP Layers and compares their suitability for heterogeneous environments with volatile path characteristics by means of path estimation and re-ordering capabilities. In order for multipath solutions to persist in environments such as 5G ATSSS, path estimation is necessary for efficient scheduling - as analysed extensively in this paper - and re-ordering is required to compensate for the different characteristics of the paths. If reordering is not supported, traffic arrives at the destination in any order and it is up to the recipient to deal with it, which violates the rule of transparency of the ATSSS service.

Most of the Link, Internet and Application Layer concepts in Table 2.1 have not prevailed. They have only been tested or used in limited scenarios or have failed to provide the basis for heterogeneous multipath environments. This is also true for the GRE bonding [9] known to be used in large Hybrid Accesss deployments [8], as its limited path estimation capabilities exclude scenarios beyond bundling of prioritized fixed access over another access. The search for a solution in these multipath concepts for the problem presented in the motivation of this work was therefore not successful. Even the special MPRTP for real-time multipath transmission of media content does not offer a solution for cost-efficient scheduling strategies.

Interesting development can be found in the area of protocols belonging to the transport layer as also confirmed by the analysis of multipath transport protocols for ATSSS in [69]. Candidates for broader deployments in the future include MPTCP, MP-DCCP and MP-QUIC. These protocols share some of the functionality with reference to Figure 1.1: sequencing, path estimation using congestion control, and means for re-ordering. What makes them interesting is that all three protocols are specified to be part of 5G ATSSS [11] (MPTCP) or are discussed in this context [15] (MP-DCCP, MP-QUIC). MPTCP is an enhancement of TCP and provides multipath capabilities to TCP services transparently.

Concept	First publication	TCP/IP Layer	Path Estimation	Re-ordering considered
LACP [45]	2008	Link		✓
PPP-MP [46]	1995	Link		✓
LWA [47]	2016	Link	✓	✓
LWIP [47]	2016	Link	✓	
GRE bonding [9]	2015	Internet	(✓) DSL sync rate	✓
ILNP [48]	2012	Internet		
ILA [49]	2015	Internet		
Shim6 [50]	2005	Internet		
PMIPv6 [51]	2006	Internet		
LISP-HA [52]	2015	Internet	✓	✓
HIP SIMA [53, 54]	2007	Internet	✓	✓(sender pre-distorted)
DIME [55]	2017	Internet		
ECCP [56]	2012	Internet		
Multilink Proxy [57]	2009	Internet	✓	(✓) Delay equalization
MP-Bonding [58]	2016	Internet	✓	✓
MPT [59, 60]	2017	Internet		✓
CMT-SCTP [61, 62]	2006	Transport	✓	✓
ETOM [63]	2012	Transport	✓	✓
MPTCP [64, 10]	2008	Transport	✓	✓
MP-DCCP [28, 13]	2019	Transport	✓	✓ part of reference impl.
MP-QUIC [65, 14]	2017	Transport	✓	✓ STREAM mode
MPRTP [66, 67]	2010	Application	✓	✓
HTTP-RR [68]	2014	Application		✓

Table 2.1 Genesis of multipath concepts across the TCP/IP-Layers and their path estimation and re-ordering capabilites

In contrast to MPTCP, which inherits the strict in-order delivery of TCP, MP-DCCP is a protocol for providing multipath transport for latency sensitive services and/or services with no or less demand on reliable delivery. Especially, when used in an encapsulation framework [40] it enables multipath transport for Link and Internet Layer traffic. Something similar can be achieved by combining various QUIC functions, such as the multipath function and the DATAGRAM mode, although the re-ordering considerations are not yet at an advanced stage.

For the concepts listed in Table 2.1, typically the scheduling algorithms are implementation specific (not standardized). However, if the goal is to develop and evaluate a new scheduling algorithm an implementation is required, and the available research demonstrates that transport layer protocols are able to provide the scheduling required path information from the CC algorithm in use. Since the multipath scheduler is expected to dominate a solution for the objectives of this work, a special analysis is performed in subsection 2.1.1.

Congestion Control (CC), as defined for TCP, as input for scheduling decision has a huge impact on the performance as it controls path usage. Information provided by the CC - e.g. path availability and latency - can be used by multipath schedulers to schedule traffic over available paths. Inaccurate information from CC will lead either to underutilization or over-consumption of paths. Both have a direct impact to the multipath aggregation performance. For the purpose of this work the detailed analysis of the CC $\leftrightarrow$ scheduler interplay is out of scope, even though the most relevant concepts are investigated in section 2.3.

2.1.1 Multipath schedulers

After analysing existing multipath concepts, which only helped to understand how they work and how they relate to ATSSS and HA, several multipath scheduling strategies are now identified and analyzed in search of a solution to mitigate the observed inefficiency of additional path costs in the transmission of VoD traffic. With focus on the cost optimized multipath video streaming objective of this work, five groups of schedulers were of special interest: *Basic*, *Reduced Head-of-line blocking*, *Video optimized*, *Cost optimized* and *Cost & Video optimized*. According to these categories, Table 2.2 lists the schedulers investigated in this section. This table also defines how closely the scheduler and the transmitted service must work together in order to achieve the respective optimisation goal.

Scheduler	Category	Service dependency
Round-Robin [70, 71]	Basic	De-coupled
Load-balancing [72]	Basic	De-coupled
Lowest-RTT-first [16]	HoL reduction	De-coupled
ECF [73]	HoL reduction	De-coupled
DAPS [74]	HoL reduction	De-coupled
Blest [75]	HoL reduction	De-coupled
OTIAS [76]	HoL reduction	De-coupled
STTF [77]	HoL reduction	De-coupled
Peekabo [78]	HoL reduction	De-coupled
Redundant [79, 80, 81]	HoL reduction	De-coupled
PO-MPTCP [82]	Video optimized	Interface
B&B [83, 84]	Video optimized	Integrated
LBA [83, 84]	Video optimized	Integrated
Cross-Layer [85]	Video optimized	Interface
Multipath Adaptive Video Streaming [86]	Video optimized	Interface
MDC [87, 88, 89]	Video optimized	Integrated
LPC [90]	Video optimized	De-coupled
LET [90]	Video optimized	De-coupled
Bandwidth probing [91]	Video optimized	De-coupled
CPF/Strict-Priority [16]	Cost optimized	De-coupled
ACPF [23]	Cost optimized	De-coupled
MP-DASH [92]	Cost & Video optimized	Interface
Pref MP-SVC [93]	Cost & Video optimized	Integrated
Pref MPTCP-SVC [93]	Cost & Video optimized	Interface

Table 2.2 Multipath schedulers categorised according to the intended use and their dependency on the service they are scheduling

Basic schedulers using round-robin method [70, 71] or load balancing [72] fail to respect cost policies which strictly prioritize path against each other.

For schedulers aiming to reduce the *Head-of-Line (HoL) blocking*, the foremost goal is to ensure a continuing traffic flow, overcoming any interruptions caused by disjoint path latencies. As [70] points out, opportunistic re-transmission is a general method to improve the responsiveness in a multipath system when it comes to packet loss. Schedulers with different strategies – minimizing the out-of-order delivery, the overall completion time, the shortest delivery or a combination thereof – to reduce HoL blocking are Lowest-RTT-first [16], ECF [73], DAPS [74], Blest [75], OTIAS [76] and STTF [77]. More information about their individual strengths and the path parameters taken into account for optimization can be found in [94]. Peekabo [78] claims to outperform the other schedulers using deterministic and stochastic methods for faster and more accurate decisions in heterogeneous and volatile environments. Another method to resolve HoL blocking is the usage of different forms of redundant transmission [81]. With the clear goal of maximizing the experience of latency sensitive services, HoL blocking optimized scheduling, however, follows an arbitrary and therefore non-deterministic logic from a path cost point of view.

While some of the above HoL blocking sensitive schedulers have also proven to be efficient for video streaming, there are specific schedulers focusing on *video optimized* scheduling. PO-MPTCP [82] enhances Lowest-RTT-first scheduling by prioritizing video traffic over non-video traffic. Other approaches extend this idea to model the multipath system and consider path parameters like throughput, latency, buffering and loss rate. Using this information helps to design scheduling logic which can dispatch video data to ensure arrival pattern necessary for smooth decoding of video at the receiving end. Such schedulers are known to work optimally when the multipath system model is accurate. [83, 84] demonstrate this for Scalable Video Coding (SVC), while [85] favours a cross layer approach using MPTCP for gathering path characteristics. Along this line, [86] resembles the idea from [85] and elaborates a coupling of MPTCP information with Dynamic Adaptive Streaming over HTTP (DASH), the commonly used method for VoD transmission in the Internet. A video coding scheme designed from ground up for path diversity is Multiple Description Coding (MDC) [87]. MDC encodes complementary descriptions from a media stream to built redundancy and therefore resilience to losses when it is requested over diverse paths. [88] states that MDC is suitable for scenarios in which no feedback loop is possible, but under conditions of limited path heterogeneity. This is where [89] provides a solution and combines MDC with a multipath model for optimized scheduling of high-weighted data,

similar to the concept outlined in the beginning of this paragraph.
Different to the approaches which integrate scheduling with the video application or the video coding, the decoupled solutions tend to optimize multipath scheduling for video-streaming independently. In that sense, [90] focuses on two different scheduling strategies for MPTCP which either prefer the path with the largest Congestion Window (`CWND`) or the one with the largest estimated throughput. In comparison to the Lowest-RTT-first scheduler, the results vary across the scenarios, with a strong dependency on the congestion control in use. [91] goes a different way and monitors the total and the per-path throughput to detect if a particular path contributes efficiently or if it is beneficial for the total throughput to stop transmission over this particular path. Similar to the *HoL blocking* scheduler, the *video optimized* schedulers does not use any path cost metric.

The task of a *cost optimized* scheduler is then to distribute traffic according to the given path costs. [16] describes the Strict Priority scheduler which consumes the available – non-blocked – paths in the order of cost. This corresponds to the default scheduler in the GRE based HA [9], denoted there as Cheapest-path-first (CPF).
Even if better QoE is possible, the impact of VoD over MPTCP with CPF logic was negatively evaluated in [21]. Due to the bursty nature of the VoD traffic, CPF is unnecessarily triggered to overflow into high cost path generating spurious demand without QoE benefit. This is also not changed by ACPF [23] which extends CPF to optimize aggregation performance when the multipath systems encapsulate congestion controlled end-to-end traffic.

A set of scheduler solutions which aim to combine both the cost efficiency and the optimized video transmission are the *cost & video optimized* ones. Authors of [92] proposed a multipath extension of DASH with a scheduler which favours the low-cost path over the high-cost path as the primary goal. This broadly follows the CPF logic, but is not as strict, as adjustments are made if the deadline for video chunks cannot be met. For implementation, the idea of [85] is resembled, using a cross-layer approach with tight integrated MPTCP and DASH video client. The concept demonstrates high levels of QoE as it profits from the *delay-tolerance* of VoD chunks as it is needed to cope with the delay dispersion imposed by the typical Internet connectivity. Different to DASH, which uses Advanced Video Coding (AVC) for adaptive bitrate streaming, an alternative exists. Scalable Video Coding (SVC) promises to reduce stall events by encoding chunks in ordered layers. Hence, with the reception of the basis layer it is already possible to playback the video and with requesting and receiving higher layers, video quality will be enhanced. SVC is used in [93] to demonstrate an optimal solution for retaining QoE while keeping the best possible defined link preference. In that [93] is similar to [83] and [84], but path costs are taken into account. The authors force an application-integrated implementation over a cross-layer approach, even if a combination with MPTCP is considered possible. The disadvantage of this approach is that it requires access to video coding information and complex algorithms to demonstrate the lowest consumption of non-preferred paths, while providing highest video quality levels without any stall events.

Following the analysis of the schedulers presented above, it can be seen that the best decision capabilities for video optimized scheduling exist when complete and accurate information about video coding, packet delivery deadlines, playback buffer, Round-Trip-Time (RTT), available throughput, and other network information is available. This is also confirmed by [95].
In line with this and the objective of this dissertation, MP-DASH [92] and MP-SVC [93] deal with the imbalance of achievable QoE and costs. Focusing on QoE, they choose the path with the lowest cost that helps to achieve a certain QoE target. Assuming that today's access networks can generally meet the QoE requirements of VoD, this leads to perfect utilization of the low-cost path and only uses an additional path if the bandwidth requirement exceeds this. However, interaction with the carried service is a basic prerequisite. In the context of this work, it can then be concluded from the literature research presented in this section that schedulers for video optimized and even cost optimized transmission exist, and can be, according to Table 2.2, categorized as:

- *Application integrated* - Multipath scheduling within the video service
- *Hybrid* - Cross-Layer approach with interfaces between video service and scheduler
- *De-coupled* - Video service independent scheduler

The schedulers falling into the first two categories – *Application integrated & Hybrid* – have full access to the parameters mentioned above for video optimized scheduling decision capabilities. On the other hand, due to their service dependency, this does not allow quick changes of the algorithm if required and moreover exclude support of intermediate multipath architectures like HA or 5G ATSSS which claim to be service transparent as can be read in more detail also in section 2.2. Respecting this also excludes those *Hybrid* solutions analyzing the video traffic and meta-data to gain insights into the video-transmission characteristics, which is rendered impossible if for example Hypertext Transfer Protocol (HTTP)-based video deployments change to use HTTP/3 over Quick UDP Internet Connections (QUIC). Thus, solutions in the context of this work has to be searched in the area of the *De-coupled*

approaches.
The research here, however, concludes that no de-coupled – transparent to the video service – solution is available for multipath scheduling of bursty data such as VoD which optimizes both QoE and cost. The relevance of this work is well founded by the existing Hybrid Access deployments and the upcoming 5G ATSSS, which are both limited due to usage of the CPF principle for cost optimization. Therefore, the focus will be in the next sections on better understanding of the root cause and the boundaries of the observed QoE and cost mismatch in HA and ATSSS and will develop countermeasures.

Although schedulers aiming to reduce HoL blocking cannot directly support the goal of this work, they at least help to understand how certain HoL blocking scenarios can be overcome, e.g., by using opportunistic retransmission [70]. For the widely used video streaming protocols based on HTTP, DASH and HLS, this is useful as they can expect retransmission-based and therefore reliable network transport and should be considered when developing a new scheduler as part of this work.

2.2 Multi-path scheduling within Hybrid Access and 3GPP ATSSS

The concept of 3GPP ATSSS, as first specified in 3GPP rel. 16 [11], defines the new feature of a Multi Access Protocol data unit (MA-PDU) session to connect a user equipment (UE, e.g. smartphone) to a data network (DN, e.g. Internet). Compared to the traditional PDU session of 3GPP, the MA-PDU has two legs – multipath. One leg is over 3GPP, and the other over a non-3GPP access, for example a Wi-Fi or a wireline access. Even with one leg missing, the MA-PDU session stays functional. For the usage of ATSSS, three operating modes are defined by the S's: Steering, Switching and Splitting. In the first two modes, a specific access is selected for transmission, with Steering as a permanent decision and Switching as a reversible decision for the affected traffic. Conclusively, this means Steering allows initial access selection, while when Switching is configured, traffic can be seamlessly shifted between the legs without interruption. However, both modes rely on single path transport. Contrary to this, the Splitting mode defines the simultaneous usage of the access legs for gaining higher throughput. This requires a per-packet multipath scheduler where ATSSS specifies the following traffic steering modes for splitting:

- Smallest delay: Prefers link with the lowest RTT.

- Load balancing: Link sharing using a specified ratio.
- Priority based: Prefers the path with higher priority. Inline with the CPF principle used in HA, see section 2.1.

Following the multipath 3GPP rel. 16 specification, the transport layer protocol covering **all** three S's is MPTCP. Sidenote: With the MPTCP implementation of CPF and the new COM scheduling algorithm as part of the presented work in this thesis, ATSSS splitting as defined in 3GPP rel. 16 is fully supported.

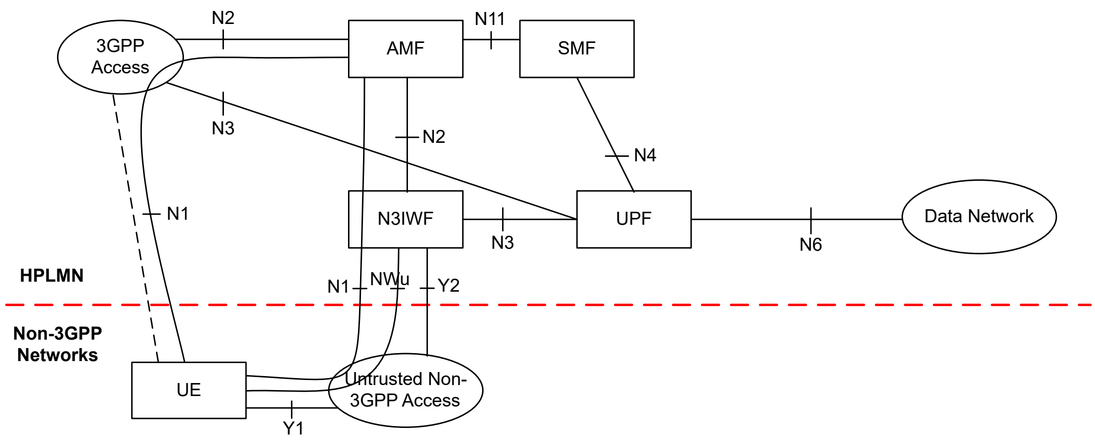


Fig. 2.1 5G system with ATSSS architecture in the non-roaming scenario and untrusted non-3GPP access [11]

In Figure 2.1 the underlying 5G system (5GS) architecture is shown. with the 3GPP numbered reference points which denote the connection between the different network entities. For the establishment and configuration of the MA-PDU session between the User Equipment (UE) and the 5G network the control plane entities Access and Mobility Management Function (AMF) and Session Management Function (SMF) are necessary including communication across the reference points N1, N2 and N11. A successful establishment of a MA-PDU session leads also to the establishment of a user plane connection between UE and User Plane Function (UPF) over the N3 reference point and connects the UE finally with the Data Network (DN, e.g. the Internet) through N6. For the configuration of the UPF resources, N4 provides a connection to the control plane of the 5G network and is for example used to configure the ATSSS settings which affect the data transmission from DN to UE. While this is the typical communication flow when 3GPP access is used, the same reference points can be used over non-3GPP access to establish a MA-PDU leg. This is achieved by the usage of the Non-3GPP InterWorking Function (N3IWF) which connects the UE over IPsec tunnel (NWu) over the reference points Y1 and Y2 through the 5G network. The latter two, for example, represent the connection of the UE to a Wi-Fi access point and from there to the N3IWF. Finally, this architecture enables the UPF to work with a single access-independent N6 IP address, making the multipath transport transparent for the DN. While Figure 2.1 shows the non-roaming (HPLMN) scenario and usage of an untrusted non-3GPP access, the two legs principle stays the same if roaming scenarios and/or trusted non-3GPP access is considered.

Between User Plane Function (UPF) and User Equipment (UE) the multipath system of ATSSS for splitting implements the components of a multipath system shown in Figure 1.1, using MPTCP for splitting. In such a system the MPTCP scheduler for traffic distribution across 3GPP and non-3GPP access resides in the UPF for traffic splitting from the DN to the UE and in the UE for traffic splitting towards the DN. For transparency towards the DN, a MPTCP proxy takes responsibility to convert TCP into MPTCP and vice versa between N6 and N3.

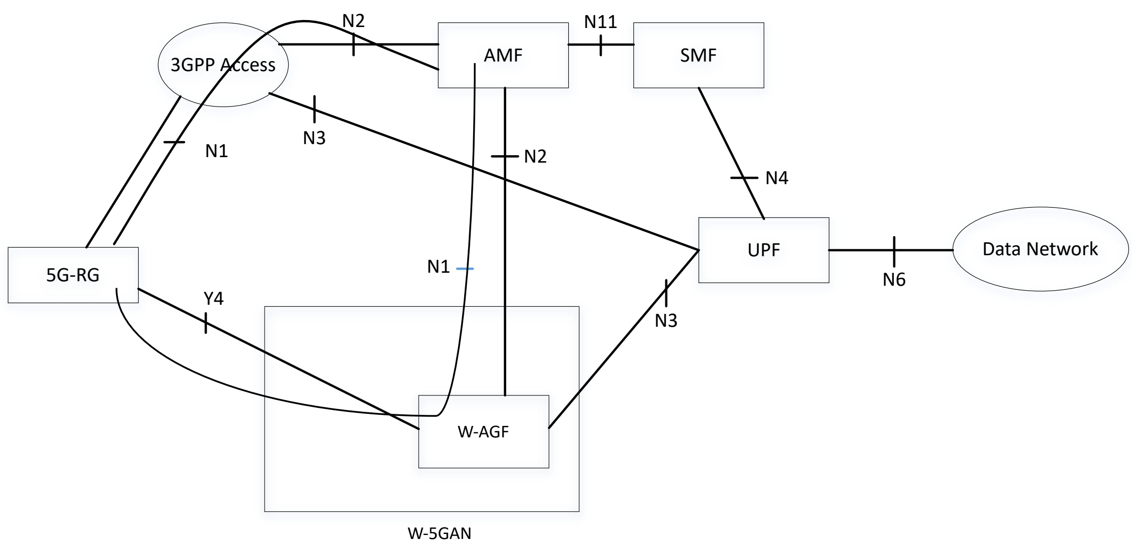


Fig. 2.2 Wireless Wireline Convergence architecture for Hybrid Access using 3GPP ATSSS in the non-roaming scenario [11]

In a quite comparable way, Hybrid Access using ATSSS is specified in 3GPP rel. 16 Wireless Wireline Convergence (WWC) [96], which supersedes [97] with tight 5G integration.. Compared to the UE scenario (Figure 2.1) the UE is replaced by a 5G Residential Gateway (RG) in HA as shown in Figure 2.2. Instead of the untrusted non-3GPP access with reference points Y1 and Y2 and N3IWF, Y4 specifies a wireline access such as DSL or FTTH with the Access Gatway Function (AGF) as entry point to the 5G core. From a multipath system perspective, no change is provided though, with the exception that a second MPTCP Proxy on the 5G-RG is defined for transparent multipath transport to 5G-RG connected devices. Another WWC exclusive feature is the support of 4th generation mobile networks (4G) and 5G for 3GPP access connectivity.

2.3 Congestion Control

In the multi-path scenario CC helps multipath protocols to exist in heterogeneous and volatile path setups. As valuable supplier of path characteristic information such as remaining capacity or RTT it is an enabler for scheduling logics as described in section 2.1. At the same time it keeps its original intent to avoid over utilization of path resources and checking the Congestion Window (CWND) is in most scheduling approaches the basis to verify path availability.

In the area of transport protocols Congestion Control (CC) play an essential role as it determines the available path throughput under operation and adapts to changing transport conditions. TCP, Datagram Congestion Control Protocol (DCCP), Stream Control Transmission Protocol (SCTP), QUIC and their multi-path pendants use it to avoid over utilization of network links and middle-boxes. The effect of over utilization would raise buffer bloat with prolonged latencies and in the worst case packet loss. CC algorithms control the path characteristic estimation and the reaction on changing path parameters such as packet reception or latency, which leads to the calculation of a Congestion Window (CWND). Typically, the CWND is the limiting transmission factor, which is reduced whenever data is sent and freed by the amount of data acknowledged by the receiver. If the CWND is empty, $CWND = 0$, transmission stops and can only be recovered by incoming acknowledgements or by Retransmission Time-Out (RTO).

It is in particular depending on the goal of the CC algorithm how to reduce the CWND when anomalies are detected and how to recover afterwards and differs, for example, between algorithms such as Cubic, Reno, etc. This has an impact to the different phases of congestion control known as *slow start*, *congestion avoidance* and *fast retransmit* described in [98]. Most often CC algorithms are further designed to keep fairness if multiple CC connections share the same bottleneck. Otherwise aggressive CCs would squeeze out conservative ones, letting them starve in the worst-case. Hence, algorithms are in particular suited for different access characteristics (e.g. radio or wired) or for example optimized for multi-path transmission, while others try to provide a trade-off to work across a broad range of different scenarios.

CC in its role as path estimator (see Figure 1.1) as used by MPTCP, MP-DCCP, MP-QUIC and Concurrent Multipath Transfer SCTP (CMT-SCTP), provides the boundaries in which a multi-path scheduler can operate. Thus, a scheduler can only work as well as accuracy of CC allows. Overestimation of path characteristic might lead to packet loss or buffer-bloat, while underestimation leaves available resources unused. In both cases efficiency is impacted. However, as the goal of this thesis is to explore optimized multi-path scheduling, CC is just considered as an available mean to support this goal.

Hence, the optimization of CC in connection with scheduling is not in scope of this work. It is, however, recommended to verify practicability of scheduling approaches in the scope of this work in case they rely on CC information.

The reasons for this statement are manifold:

1. This is seen in line with the requirement of this thesis, that the research regarding optimized multipath scheduler has no dependency to a particular multipath network protocol and therefore can be implemented in MPTCP – uses CC – but can also be implemented in a different protocol like GRE based HA – does not use CC.
2. If CC provides relevant input to the multipath scheduler it is enough to prove that across different CCs the trend of results persists.
3. The target scenarios ATSSS and HA typically have no shared bottleneck or if so, for example in the case of Wi-Fi over N3IWF (section 2.2) or the cellular link itself, tend to have per-user queues on lower layers.
4. Even if multipath optimized CC exists, leading one to believe that it improves multipath transport, its focus would be on fairness, preventing the combined throughput of a multipath connection from taking up more than its fair share over potentially shared bottleneck links. This only impacts the ratio of consumable resources across the links but does not change the principal scheduling logic.

Nevertheless it is worthwhile to understand how the CC landscape looks like in order to classify the above statements. In the category of CCs which provide a compromise between efficiency, fairness and universality, *Cubic* [99] and *Bottleneck Bandwidth and Round-trip (BBR)* [100] have evolved. They present the most dominant CCs in the end-to-end Internet communication today, and therefore seem to be the most valuable for testing, as they have already proved applicability with VoD traffic. In the following sections, *Cubic* and *BBR* are briefly introduced, as well as multi-path optimized CC. The latter category has only informational character as no CC based multi-path deployment is known. As their main focus is on shifting away traffic from bottlenecks shared with single-path transmissions, no surprising results are expected in performance comparison using *Cubic* and *BBR*. The only possible impact is a lower availability of a cost efficient path in the case unfairness on this path is observed.

2.3.1 Cubic

Cubic, with its packet loss sensitivity, has evolved as default CC on Linux since 2006 [101], Mac OS since 2014 [102] and Windows & Windows Server since 2017 [103]. This means, *Cubic* is always in use unless default setting is manually overwritten by changing the system wide default or selecting specific CC when a communication socket is created using socket option[1]. The main characteristic, given by the name, is a cubic function determining the CWND growth based on packet loss events. As packet loss events leads to a CWND reduction, the last measured CWND, W_{max}, before such a reduction holds as inflection point where a concave growth transitions to a convex growth, see Figure 2.3. The concave growth ensures especially after CWND reduction a fast recovery. The typical sawtooth pattern known from packet loss based congestion control also exists in *Cubic*, which is caused by maximum probing and consequently leads to packet loss.

2.3.2 BBR

BBR, unlike Cubic, is not an operating system (OS)) standard CC, but is widely used because it is available in the Alphabet universe [104, 105] for the Google search engine, data centre communications and Youtube, and as public code for Linux TCP and QUIC [100, Sec. 5]. The optimal operation point *BBR* aims to achieve is where maximum throughput at minimum latency can be achieved. As this goal is persistent across version 1 and 2 of *BBR*, in the latter the latency dependency is complemented to take packet loss into account for better fairness with other CCs. In Figure 2.4 the *BBR* operation point is drawn over the bytes in flight separated by throughput and RTT development. Using Bandwidth Delay Product (BDP) to denote this optimal operation point, shows the effect on intermediary network buffer occupancy which is avoided. Consequently, no additional latency is introduced when throughput is paced at maximum level without overflowing real path capacities without buffers. Another side effect of low RTTs can be expected in multi-path systems. Usually

[1] https://man7.org/linux/man-pages/man7/tcp.7.html

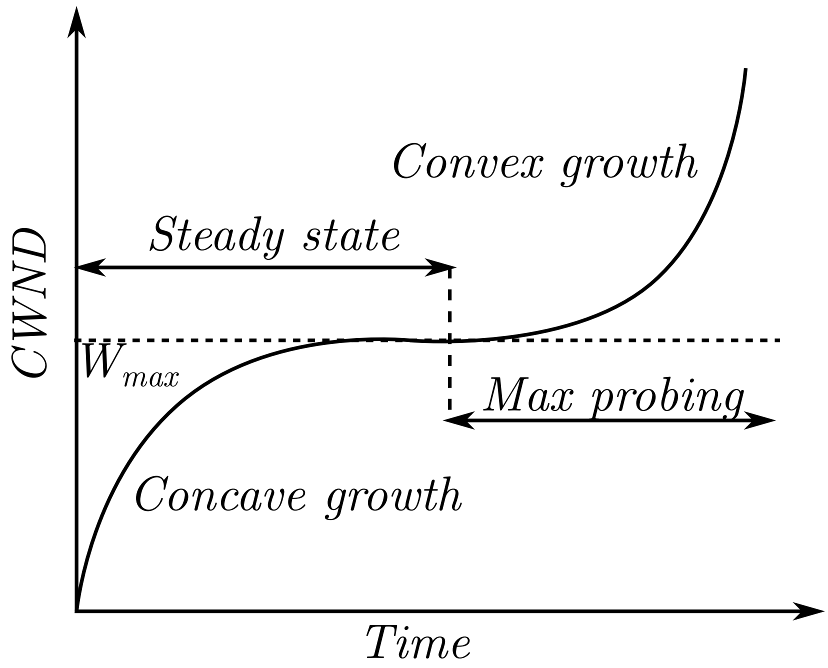


Fig. 2.3 Cubic congestion window development over time

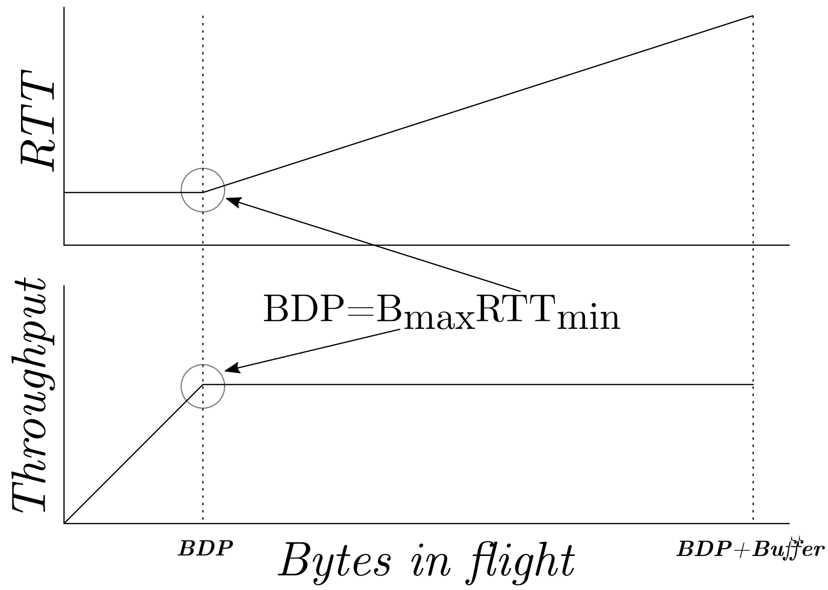


Fig. 2.4 BBR optimal operation point at BDP with maximum throughput and lowest round trip time

re-ordering of scheduled data has to compensate for the path latency differences. With increasing latency differences the re-ordering process becomes more cumbersome as more data has to be stored in the re-ordering buffer. From this perspective, it is advantageous to avoid long latencies as introduced by pure packet loss triggered CCs. Unlike the packet loss-based CC with the sawtooth pattern, BRR typically provides a constant pacing rate according to the estimated BDP.

2.3.3 Multi-path optimized congestion control

Optimized congestion control for multi-path transport developed mainly from the need to improve throughput and to ensure fairness to concurrent congestion controlled single-path flows, which is in particular important at shared bottlenecks [106]. To achieve this individual goals, *uncoupled* and *coupled* approaches exist [107]. The difference lies in the management of the congestion window, which is either treated separately by subflow – *uncoupled* – or by control of a total congestion window across all subflows – *coupled*.

Due to the development of MPTCP, coupled congestion control has prevailed in standardization and implementation known as LIA[2] [108], OLIA[3] [109], BALIA[4] [110] and wVegas[5] [111].

In multi-path deployments, where the fairness design goal doesn't play a major role, slightly better throughput performance is achievable using single-path – *uncoupled* – CC like Cubic [112].

2.4 VoD transmission principle

To understand the critical path capabilities when it comes to VoD transmission it is necessary to shed light on the VoD implementation and the resulting demands. The findings of this section are crucial to understanding why, on the one hand, VoD traffic is problematic for cost-optimised multipath transmission, but on the other hand, the section also lays the foundation for a solution elaborated later in section 3.5.

VoD transmission, as outlined in Figure 2.5, takes place between the entity which provides the video for playback - Server - and the entity which requests the video for playback - Client. Since a video is a continued playback of images - frames - a full video consists of a number

[2] https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.95/net/mptcp/mptcp_coupled.c
[3] https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.95/net/mptcp/mptcp_olia.c
[4] https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.95/net/mptcp/mptcp_balia.c
[5] https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.95/net/mptcp/mptcp_wvegas.c

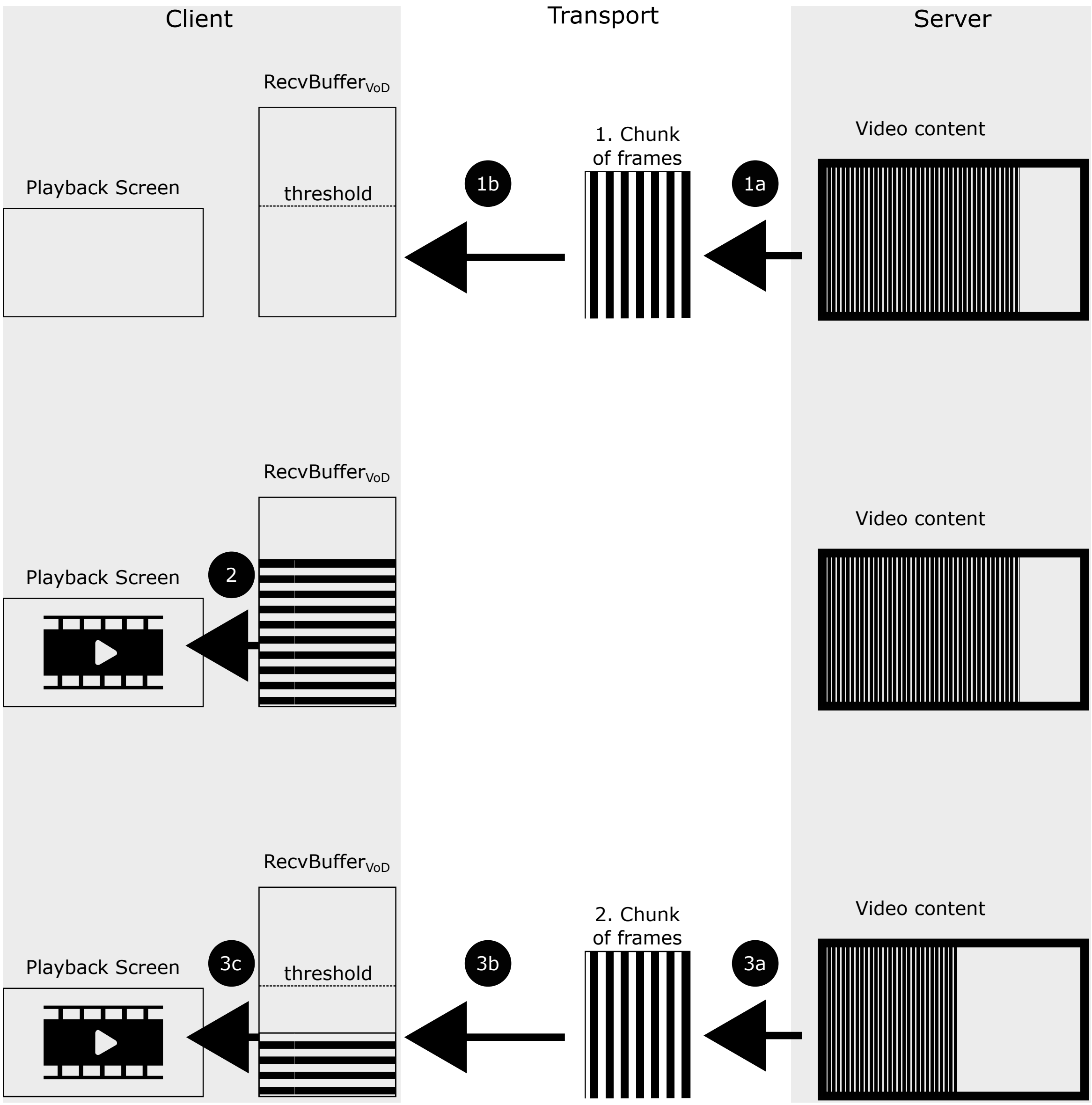


Fig. 2.5 VoD principle

of individual video frames. Usually the resolution r of the frames and the playback speed s determine the demanded throughput, B_{demand}. For uninterrupted playback at the receiver, an essential criterion is that the bandwidth of the transmission path B_p is not smaller than B_{demand}. Compression methods like H.264 [113], H.265 [114], VP8/9 [115, 116], however, help to minimize the data footprint of frames or a row of frames which results in a lower B_{demand}. In the initial phase of a video transmission, the Client requests a video from the Server. In that moment, the Server starts to dispatch video frames - Step 1a. On the Client side a receive buffer $RecvBuffer_{VoD}$ stores the received frames - Step 1b - and will forward the frames to the playback application when a certain buffer level - application dependent - is reached - Step 2. The time until the first video frame is finally presented to the consumer after requesting is denoted as $t_{initial}$. When the number of stored frames in the $RecvBuffer_{VoD}$ falls below a certain threshold, the next chunk of frames, ch, is requested and Step 3 recalls the previous steps. The threshold value can be a dynamic value, for example based on r or B_p. The concept of receiver buffering is introduced to overcome variations in path characteristics such as throughput, latency or jitter and provide a smooth playback experience. The playout of VoD content can be generally distinguished between **static** and **adaptive**. In the first case the video resolution r corresponds to a fixed value, while for the adaptive r is dynamically selected and for example reduced if a smooth playback experience is not ensured anymore.

A standardized solution which facilitates the playout of static and adaptive VoD content is the Dynamic Adaptive Streaming over HTTP [117] also known as MPEG-DASH. Its usage is widespread across the VoD content providers like the popular Youtube, Netflix, Amazon and content delivery networks like Cloudflare, Akamai, Fastly, Microsoft Azure, which all supports the DASH, incorporating the principle from Figure 2.5. A similar approach is the HTTP Live Streaming [118] developed by Apple, but limited to H.264/265 encoded material. Modern Internet Browsers or VoD applications and other client software, as well as operating system support typically at least one of these standardized transmission methods. What unites them all, is the segmentation of full length video into small pieces of seconds and provide them with different encoding schemes according to video resolution(s) r (... , 240p, 380p, ... , 1080p, ...). Based on a client logic (static or adaptive), the next segment of the video with a selected r is requested over HTTP, which is then transferred by the Server as a chunk to the Client.

The VoD traffic pattern tp_{VoD} is shaped by the dispatched chunks ch_j, where $j = 1, ...M$, with j denoting the individual chunk in the video segmentation range M $\{j, M \in \mathbb{N}\}$. A chunk ch_j can be characterized by the chunk size in terms of volume $V_{ch,j}$ in relation to the chunk duration in time, $t_{ch,j}$ and is depicted in the "Generation" column on the right of Figure 2.6. The duration of a chunk is further limited by a starting and ending time $t_{ch\_start,j}$ and $t_{ch\_end,j}$. Depending on the path transport capacity B_p, the generated chunk shape is received unchanged – $T_{GAPrecv} = T_{GAPsend}$ – or undergoes transformation, which let the time interval

$$T_{GAPsend} = t_{ch\_start,j+1} - t_{ch\_end,j} \tag{2.1}$$

between consecutive chunks melt – $T_{GAPrecv} < T_{GAPsend}$. This means a bottleneck in the transport path, after generating and dispatching the chunks burst-wise, leads from a bursty pattern in Figure 2.6a and 2.6b to a more and more compressed burst, up to the point where consecutive bursts merge and fill the pipe (Figure 2.6c). However, if the bottleneck is too small, the bursts even overlap and this causes packet loss as seen in Figure 2.6d. In an adaptive streaming scenario, the client logic should then request video segments with lower r. The transformation discussed here is independent of where in the path the bottleneck occurs, however it can be imagined, that often the last mile towards the consumer acts as the bottleneck link within the end-to-end transport path.

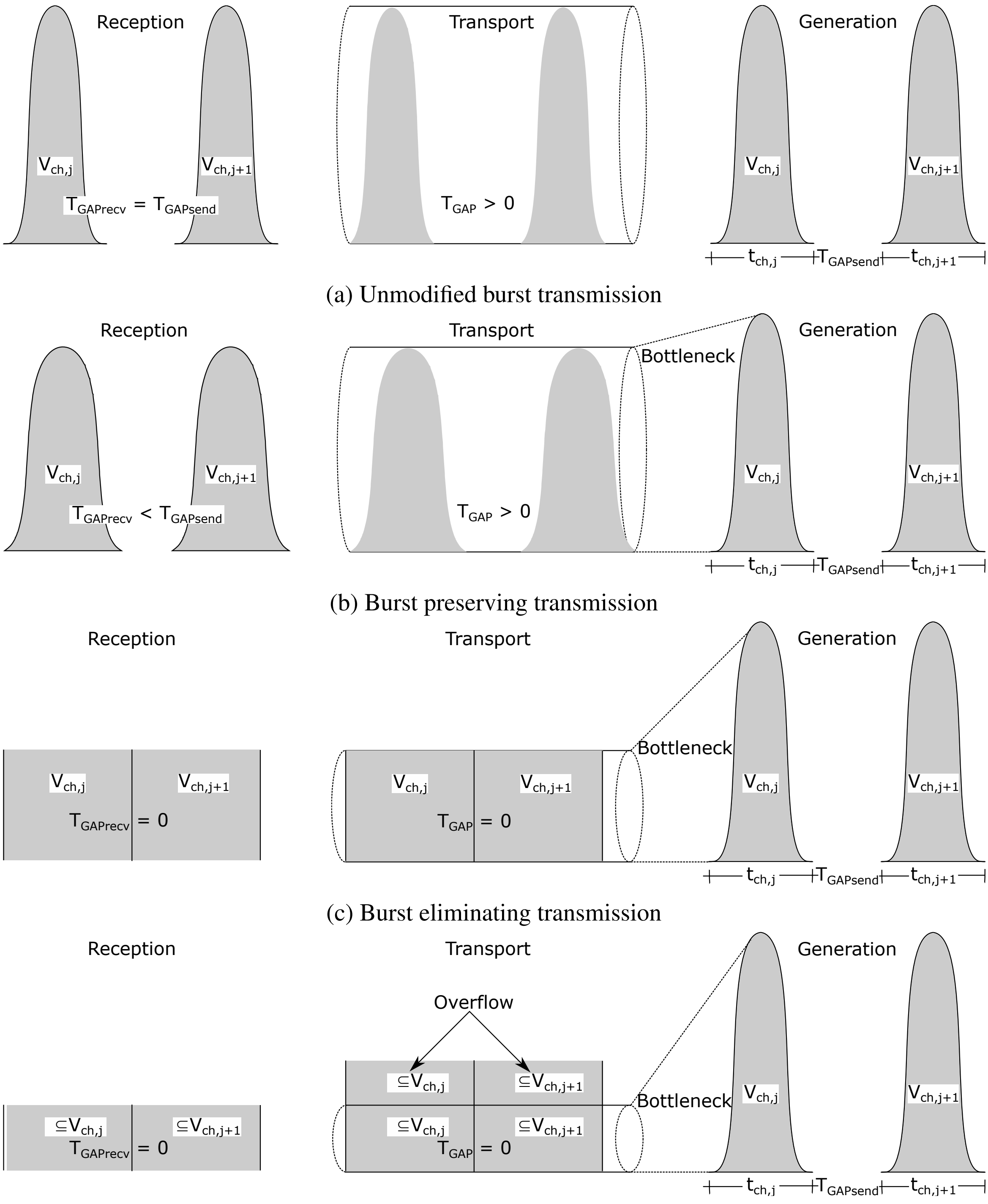
$$tp_{VoD}(ch_j) = \begin{cases} Burst & \text{if } B_p > \frac{V_{ch,j}}{t_{ch,j}+T_{GAPsend}} \\ Fill & \text{if } B_p = \frac{V_{ch,j}}{t_{ch,j}+T_{GAPsend}} \\ Overflow & \text{if } B_p < \frac{V_{ch,j}}{t_{ch,j}+T_{GAPsend}} \end{cases} \tag{2.2}$$

Typically sending the chunks as fast as possible from the Server to the Client is in the interest of the VoD provider to keep the playback process ongoing. This is also reflected in the selected transport protocols. According to DASH and HTTP Live Streaming (HLS), this is HTTP transmission over congestion controlled protocols like TCP or QUIC which tend to operate at the maximum throughput level given by the bottleneck in the end-to-end transport chain. As explained in Figure 2.6, consecutive chunk bursts are generated with an inter-dispatch time $T_{GAPsend}$, which is in case of the "Burst" or "Fill" case greater or equal than the received inter-arrival time $T_{GAPrecv}$. This is further outlined in Equation 2.2, which emphasizes again the dependency on the tranmission path throughput B_p. Effectively in the first two cases, the chunk burst gets stretched or stays even, while for the "Overflow" case the bottleneck distortion let the burst compress and $T_{GAPrecv}$ becomes smaller or cannot be measured anymore. However, as long as the transmission operation point falls into the "burst" or "fill" case, the video can be enjoyed without limitations as long as the $RecvBuffer_{VoD}$ does not run out of video frames, while in the overflow case the video transmission definitely stall for a certain time until new valid video frames are received, this time is denoted as t_{stall}.

Two things to remember after reading this section:

1. **If the last mile in the transport chain shapes the generated VoD traffic pattern, the impact becomes less relevant if the bandwidth of the last mile increases, as is the aim of HA or ATSSS concepts.**

2. **If the receiver buffer $RecvBuffer_{VoD}$ is designed to compensate for the transport characteristics, smooth playback is guaranteed in the *Burst* and *Fill* cases defined in Equation 2.2.**



(a) Unmodified burst transmission

(b) Burst preserving transmission

(c) Burst eliminating transmission

(d) Burst eliminating and lossy transmission

Fig. 2.6 VoD transmission traffic pattern

2.5 VoD QoE measurement

With the Mean Opinion Score (MOS) a measure exists to express the QoE for video and audio content in a typical range between 1 (bad) - 5 (excellent). In [119], [120], [121], [122] and [123], different models for measuring a MOS for VoD QoE estimation are discussed. They all focus on the particular model in [124], which is represented as a block diagram in Figure 2.7. It is referring to QoE estimation of packets based VoD transmission over reliable transport protocols (TCP, QUIC) and helps to understand relevant input parameters and outputs of QoE MOS determination. In more detail [124] is applicable for:

- Progressive download streaming and adaptive streaming (using reliable transport), which includes:
  - Over the top (OTT) services, as well as operator managed video services (over TCP).
  - Video over both mobile and fixed connections.
  - The network protocols HTTP/TCP/IP, RTMP/TCP/IP, HLS/HTTP/TCP/IP, and DASH/HTTP/TCP/IP. Note that the model is agnostic to the specific network delivery method (HTTP or DASH or other), with one exception that it assumes reliable delivery (TCP/IP).
  - Video services typically using container formats such as Flash (FLV), MP4, WebM, 3GP, and MPEG2-TS. Note that the model is agnostic to the type of container format.

While the final MOS value builds an integral value over audio quality, video quality, device parameters and stalling events, the input parameters I.GEN, I.11, I.13 and I.14 therefore distribute as follows in Table 2.3, 2.4, 2.5, 2.6 with uncolored cells referring to parameters affecting the whole media sequence, light-gray cells covering segments with same quality level (used for chunk delivery, see section 2.4) and dark-gray cells consider the most granular frame unit.

I.GEN	
The resolution of the image displayed to the user	The device type on which the media is played

Table 2.3 Generic input parameters I.GEN ITU-T P.1203

Given by all this parameters and the comprehensive model with its individual calculation described in [125, 126, 127], the final calculation of a comparable MOS value becomes quite

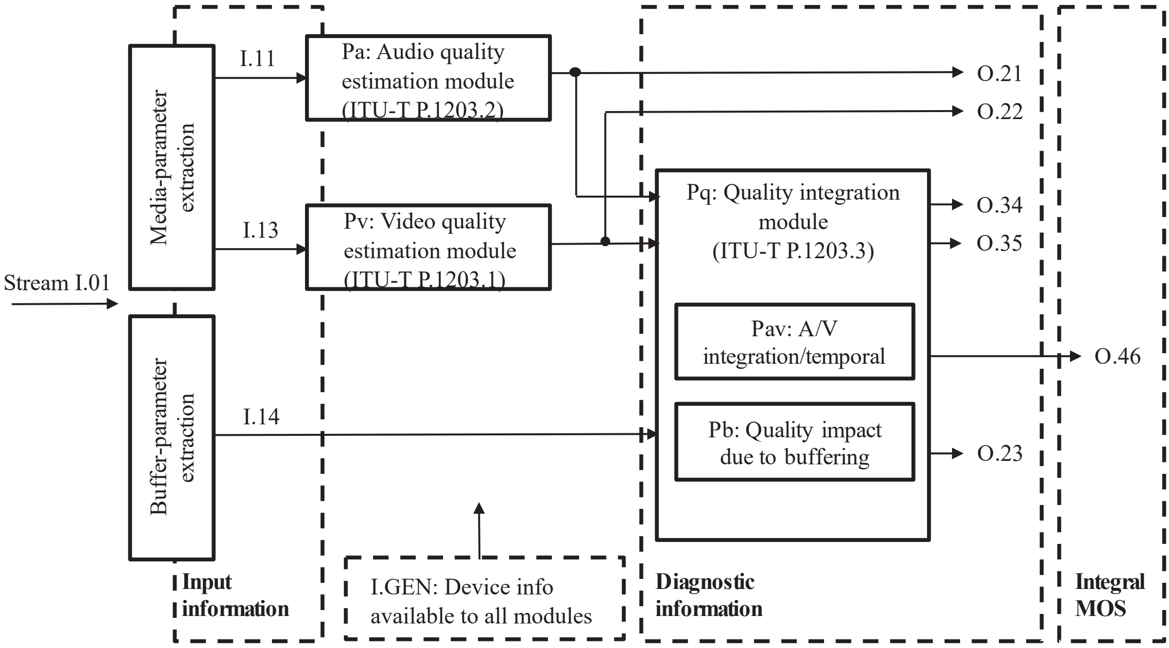


Fig. 2.7 Building blocks of the ITU-T P.1203 VoD QoE model

I.11		
Target Audio bit-rate	Segment duration	Audio frame number
Audio frame size	Audio frame duration	Audio codec
Audio sampling frequency	Number of audio channels	Audio bit-stream

Table 2.4 Audio input parameters I.11 ITU-T P.1203

I.13		
Target Video bit-rate	Video frame-rate	Segment duration
Video encoding resolution	Video codec and profile	Video frame number
Video frame duration	Frame presentation times-tamp	Frame decoding timestamp
Video frame size	Type of each picture	Video bit-stream

Table 2.5 Video input parameters I.13 ITU-T P.1203

I.14	
Stalling/initial loading event start	Event duration

Table 2.6 Stalling event input parameters I.14 ITU-T P.1203

laborious. However, the model fits in the scope of this work to investigate the impact of multipath scheduling over typical VoD transmission, as provided for example by Youtube and other VoD provider. Considering – for example – the determination of the video related MOS calculations in [125], the played out video resolution and the stalling times play the dominating role, as long as the frame rate is stable. It will be presented in section 3.4 that the VoD QoE determination over different network transport characteristics can be simplified and limitations such as the model validity for H.264 codec, a frame rate between 8-30 a maximum video resolution of Full HD and so on as outlined in [124, Table 1] is negligible.

Chapter 3

System model, Problem Statement and Solution

In the section 1.1 a cost contradiction in an commercial access aggregation scenario is disclosed where in Figure 1.2 a significant traffic overflow from a cheaper path into an expensive path is triggered when VoD is transmitted. It is the goal of this chapter to gain evidence that this is a natural consequence of the combination of access aggregation with a cost preference and the transmission of services with VoD traffic pattern. The second goal of this chapter is to develop a strategy to mitigate this effect, especially under the recognition that the aggregation of an expensive path is not equivalent to a better QoE.

To reproduce the problem in a first step, a multipath system model which considers cost is described in section 3.1. After discussing the relevance of multipath scheduler which prefer low cost path over high cost path in section 3.2, this model is further used to develop a path cost driven prioritization scheduler called *Cheapest-path-first (CPF)* for the relevant multipath network protocol MPTCP.

In subsection 3.2.5, it is possible to reproduce the problem indicated above showing a huge discrepancy between caused transmission cost and achieved QoE in the case of the dominating VoD traffic. This leads finally in section 3.3 to the formulation of the objective of this work.

A solution for this problem is developed in section 3.5 based on traffic engineering leveraging the delay tolerant property of VoD services with its main dependency on throughput as identified in section 3.4. Within an iterative process a new scheduling algorithm *COM* is designed and also implemented for MPTCP as substitute to *CPF*. The guiding principles developed in this process focus on maintaining compatibility with the target architecture of

ATSSS and HA (section 3.8) and avoiding impacts to traffic that has a different pattern than VoD (section 3.7).

3.1 System model

A general case of a multipath access network is observed as depicted in Figure 3.1. The network model consists of two termination points and N distinct paths denoted p_i, where $i = 1, ...N$, with i denoting the individual path in the range of available paths N $\{i, N \in \mathbb{N} | N \geq 2\}$. A system with the minimum number of paths therefore consists of p_1 and p_2. All paths together form a composite multipath connection *mc*. The data, as a stream of PDUs, enter the *mc* and is split at the first termination point into multiple paths p_i, according to a scheduling logic. At the other termination point, the split traffic from the paths p_i is aggregated again and forwarded. The multipath system is characterized by giving access to the paths p_i simultaneously. This allows fine granular scheduling on the PDU level and is therefore suitable to provide an aggregated experience of the individual path characteristics within the *mc*.

The typical path transmission characteristics include: the amount of data B, which can be transmitted according to the path throughput, the path latency L, the latency variation - Jitter - J and the loss-rate of data units R. While this definition of transmission characteristics applies to both the individual p_i and the composite *mc*, a resource efficient path selection process for traffic splitting (scheduling) is dependent on an additional characteristic, the path cost C. This work assumes that the path costs are a predetermined metric, e.g. set by the operator of the multi-access network.

In a resource efficient multipath system, C_i assigns therefore a cost value to each p_i with an increasing C indicating a higher cost. The determination of the path cost feature could be for example based on the actual monetary cost or on some latency-related criteria.

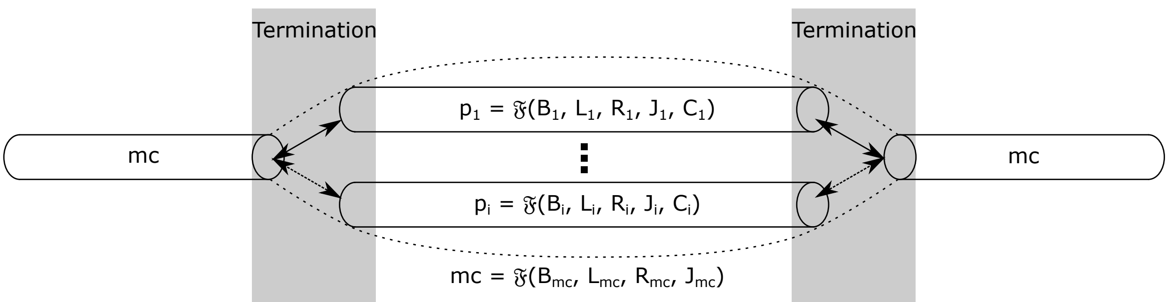


Fig. 3.1 Network model of a multipath-system

Then, the individual paths can be characterized as a function $\mathfrak{F}()$ of the above parameters, including the cost feature. $\mathfrak{F}()$ is used throughout this section as generic expression to denote all possible functions.

$$p_i = \mathfrak{F}(B_i,\ L_i,\ R_i,\ J_i,\ C_i) \tag{3.1}$$

whereas for the composite path, the transmission characterization is derived from the set of multiple p_i.

$$mc = \mathfrak{F}(B_{mc},\ L_{mc},\ R_{mc},\ J_{mc}, C_{mc}) \tag{3.2}$$

As an example, in a two-path system with cost parameters $p_1, C_1 = 0$ and $p_2, C_2 = 1$ a focus would be on cost efficiency, and would result in a scheduling logic that enforces usage of path p_1, with path p_2 (the secondary path) used only when the demand exceeds the first path capabilities. For this it is necessary to obtain path congestion status under operation, in particular for volatile environments.

From an aggregated *mc* perspective, the individual path characteristics contribute differently. For example, the maximum achievable throughput B_{mc}, which determines B_p – if no other bottleneck exists – and thus the resulting traffic pattern such as tp_{VoD} for the transmission of an VoD service (section 2.4), calculates as

$$B_{mc} = \mathfrak{F}(B_1,\ \ldots\ , B_i) = \sum_{i}^{N} B_i \tag{3.3}$$

and the overall cost C_{mc} is proportional to the **utilized** B_i and B_{mc} denoted as B_{Ui} and B_{Umc}

$$C_{mc} = \mathfrak{F}([C_1, B_{U1}],\ [\ldots,\ldots]\ , [C_i, B_{Ui}]) = \sum_{i}^{N} \frac{B_{Ui}}{B_{Umc}} C_i \tag{3.4}$$

On the other hand, the composite latency L_{mc}, loss-rate R_{mc} and jitter J_{mc} cannot be determined by simple addition, but by a non-linear function, as they depend on multiple factors. In section 3.4 and section 3.6 it is discussed that the parameters leading to non-linear dependencies play a negligible role due to the VoD design with playout buffers which is able to compensate typical access latencies and it will become clear that Equation 3.3 is relevant for the QoE of VoD transmission and Equation 3.4 to assess its cost. Using this general model, one can then define the optimisation objective of any scheduling algorithm. In the case of cost-effective multipath scheduling, the optimisation objective can be defined as: *design a scheduling function to distribute PDUs on available paths to minimise the overall cost* C_{mc}. As this is a complex multi-variable optimization problem, the rest of this chapter presents a heuristic solution designed based on measurements and empirical observations, and will then analyse in detail its performance.

3.2 CPF use-case, principle and challenge

The focus of this work is to consider a cost metric when dealing with traffic distribution in multipath networks. For that, section 3.1 describes a multipath system model which extends the typical path characteristics capacity, latency, loss, jitter with a path cost information C_i. Having this information available allows implementation of a multipath scheduling logic which ensures a prioritized usage of cheaper path resources and is denoted as Cheapest-path-first (CPF). The demand for this approach is further analyzed in subsection 3.2.1 and in subsection 3.2.2 it is outlined in which multipath frameworks CPF is already specified and used. An initial implementation of CPF using MPTCP is the subject of subsection 3.2.3, and basic tests performed in subsection 3.2.4 show predictable path cost results over the default latency-based scheduling shipped with the MPTCP Linux reference implementation. The large discrepancy in the spent path cost for transmitting VoD services using CPF without making a QoE gain compared to transmitting the same service over a single path, identified in subsection 3.2.5, is the cause of this work.

3.2.1 CPF motivation

The Cheapest-path-first (CPF) scheduling is based on the setting of access path preferences. In the context of multi-connectivity this is the approach to utilize one access interface to its maximum capacity before offloading data onto other interfaces to improve delivered throughput. This approach has a greater significance to the network operators than to the customers, assuming customers agreed to pay for the volume unlimited transport service, irrespective of the network in use as predicted for a 2030 scenario in [128]. The use of cellular air frequencies and radio equipment is much more expensive than the use of fixed access and Wi-Fi, and hence, network operators would like to keep the cellular network as congestion free as possible. However, until the 2030 scenario is realised and customers of especially mobile Internet providers are predominantly offered volume limited access tariffs, multi-path implementations **must** avoid an uncontrolled utilization of expensive paths such as cellular avoid conflict with customers data plans. This would be the case if the MPTCP Linux reference implementation is deployed. There, the *default scheduler*[1] provides latency based access path preference using the *minRTT* principle. Hence, it cannot ensure that cellular

[1] https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.95/net/mptcp/mptcp_sched.c

networks are used as seldom as possible. This lies in the latency sensitive nature of the MPTCP *default scheduler*, which prefers the links which guarantee low-delay transmission. For operators or customers with cost pressure, as outlined above, this is not predictable nor desirable. In the worst case, this leads to a preferred selection of the cellular link, as with the introduction of 5G networks latency is significantly reduced. An available Wi-Fi link would then possibly not used even if the Wi-Fi would be able to satisfy the demand fully. In almost any case today's latencies experienced over fixed and mobile access technologies are quite acceptable for most services and the selection of Wi-Fi or cellular (smartphone) or fixed and cellular (HA) doesn't matter, except very latency critical ones. This could be for example gaming use cases or teleoperation services. Due to further evolution of access technologies a latency based selection of access technologies will continue to diminish. Also the later discussion on the relevant QoE parameters for VoD in section 3.4 will outline, that the throughput plays the significant role in the majority of cases.

It is therefore more decisive for customers and services and finally for the QoE, that the goodput – service usable throughput – demand can be satisfied independently from the actual selected access.

At least the rule of thumb which one might think of, that a (Wi-Fi over) fixed access provides lower latency than cellular access, is not a guarantee for a CPF like behavior when the *minRTT* scheduler is used.

3.2.2 CPF in the light of 5G multi-access network architecture

Today's most widespread multipath deployments for access aggregation are Hybrid Access (HA) which combines 4G/5G access and DSL/fiber access and traditionally based on [97] with GRE [9] as transport protocol to cover the whole customer traffic mix on the Internet Layer or MPTCP [10] specifically for TCP services. [97] specifies the Least-Cost-First traffic distribution scheme amognst others which corresponds to the CPF principle. When the GRE protocol is used, the HA deployments are limited to the CPF principle, as there is no mean to gain path characteristics beyond the known DSL sync rate. However, that is sufficient, as this is the desired effect by the operators. Prioritizing the fixed access over the cellular access is desired for two reasons. On the one hand, the fixed access path has lower transmission cost compared to the cellular. On the other hand, in practice only paths with known path throughputs can be prioritized over others. As outlined, operators providing HA usually know only the static characteristic of the fixed access making use of the modem synchronization values when such a link is established. For the specific HA use case under the conditions outlined above, this does not necessarily require a typical multi-path protocol as defined for the transport layer with MPTCP, MP-DCCP or MP-QUIC. As no path throughput estimation is needed, which is one of the essential features of the transport layer solutions, any type of traffic on the Link Layer or higher can be split without limitations. Compared to transport layer multi-path protocols, this is a clear benefit as those can only serve their single-path pendants: MPTCP→TCP; MP-DCCP→DCCP; MP-QUIC→QUIC. One of those HA deployments and the largest known world-wide deployment is by Deutsche Telekom, based on GRE using additional signaling from [9, sec. 5.2.10, 5.2.11] to advertise DSL synchronization rates. In this solution, the negotiated modem synchronization rate is exposed to the scheduling entities. For a scheduling mechanism in this scenario, the data can be scheduled into the fixed access path until the known path throughput is exhausted, before overflowing into the cellular path. A drawback of this lightweight approach is that the prioritization of the cellular over the fixed access will not work or bottlenecks in the access network beyond the last mile are not recognized. Regardless of this drawback, the implementation's scheduling mechanism behaves like the CPF scheduler which iterates over a number of paths and schedules data into the path which provides throughput at lowest cost. From the Deutsche Telekom deployment it is known, that for some service/traffic types the consumption of the costly cellular path is significant compared to others, as presented in Figure 1.2.

In an updated architecture BBF defined the 3GPP ATSSS multipath framework as part of a 5G System to be used for HA, which is specified in [12]. MPTCP is the remaining protocol in this specification to achieve aggregated throughput across multiple accesses. In the 3GPP terminology, the scheduling algorithm is denoted as *Steering Mode*. For ATSSS, the specified *Priority based* Steering Mode resembles the CPF principle. The reliance on MPTCP in ATSSS makes sense when one understood that ATSSS is also used for UE multipath communication. In this scenario, if Wi-Fi and 5G is combined and no capacity information like for DSL is available, the TCP inherent path measurement (CC) helps the scheduler to determine under operation if the selected access path is available for transmission or exhausted. A similar phenomena as described for GRE multipath above was observed in [21] using MPTCP, which reports that the CPF principle - also in a HA scenario bundling Wi-Fi and Long Term Evolution (LTE) - does not provide a cost-efficient transmission experience either. Hence, this further outlines the relevance to investigate the root of QoE unsubstantiated over-consumption in cost sensitive multi-path systems. In both cases, traditional GRE HA as well as the usage of MPTCP, at least VoD was identified as a trigger.

3.2.3 CPF implementation in MPTCP

To investigate the recognized imbalance of CPF and bursty traffic it is of benefit to have a test environment at hand for further exploration. In both multipath scenarios with GRE and MPTCP as network protocols over-consumption of expensive network resources were monitored. Using one of these protocols in a test environment helps to confirm this behavior, of course, and on the other hand is the basis for finding and verifying a solution. For this work, MPTCP was chosen because it is available as an open source solution, unlike GRE HA. Of particular interest is the Linux reference implementation[2], which includes path management functionality and multiple schedulers, and is mostly the implementation used in the literature examined in section 2.1 about multipath scheduling. In a first step now CPF is implemented.

In principle, the scheduler in MPTCP has to decide which of the TCP flows accumulated within the MPTCP connection, data will be delivered over. In the MPTCP terminology those TCP flows which are assigned to an access/network interface are referred to as subflows or paths. On a closer look into the *default scheduler*, included in the MPTCP reference implementation a path cost logic is already implemented. Different to the CPF logic, however, the path latency (more precisely the smoothed Round-Trip-Time (SRTT)) is used as path metric. For the implementation of CPF, the *default scheduler* seems to be a good basis for modification, having also the advantage that optimization functions such as opportunistic retransmission are inherited for reduced HoL (section 2.1).

After analysing the *default scheduler* code an identifying the crucial decision logic, above statement about the already available path cost awareness becomes much clearer. Listing 3.1 references the part of code in the *default scheduler*[3], which is iteratively called for each available subflow when a new TCP segment is ready to be dispatched and to keep the socket `sk` in `bestsk`, which is related to the subflow with the lowest `srtt`, via which the segment can be sent. The subflow availability is given if the TCP send window `snd_wnd` indicates

[2]https://multipath-tcp.org/
[3]https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.95/net/mptcp/mptcp_sched.c#L195-L204

space for sending segments. For that, the `snd_wnd` is calculated using Equation 3.5.

$$snd\_wnd = min(snd\_buffer, \; cwnd - bytes\_in\_flight, \; rcv\_wnd) \tag{3.5}$$

This determines the minimum across the local send buffer `snd_buffer`, the receive buffer exposed by `rec_wnd` and the estimated congestion window `cwnd` provided by the TCP congestion control minus the *bytes in flight*. For each subflow, as long as this results in a value greater than 0, the subflow is considered as available.

Using `min_srtt` as variable to keep the minimum SRTT information determined so far in the iterative process mentioned above, allows to match against the current evaluated subflow SRTT `tp->srtt_us`. In the case the current SRTT is lower than the one provided in `min_srtt`, this will override `min_srtt` with `tp->srtt_us` and store the related subflow socket `sk` in `bestsk`. It has to be noted, that before the actual loop over subflows is entered, `min_srtt` has to be initialized with a maximum value, otherwise the detection of smaller path SRTT values is not guaranteed.

```
if (tp->srtt_us < min_srtt) {

   min_srtt = tp->srtt_us;
   bestsk = sk;
}
```

Listing 3.1 MPTCP path selection logic within *default scheduler* main loop

The *default scheduler* logic can be modified to implement the CPF logic. While the first relies on the SRTT information, a provided value in the Linux TCP implementation exposed by CC, a cost indicator needed for CPF is not provided. Obviously, a transport protocol is not supposed to detect cost metrics and therefore this information must be provided from the outside. In the 3GPP ATSSS, the HA or the end-to-end scenario, that is an information which can be stored along with the network interfaces available for connecting to the different access types. In a smartphone these access types are typically the cellular and the Wi-Fi interface, for a Customer Premises Equipment (CPE) the cellular and the fixed interfaces, e.g., a DSL interface.

In Algorithm 1 a generic description of a CPF logic is provided, which is executed before data is transmitted to select the path with the lowest cost. A linear traversal logic identifies the path p_i with the lowest cost C_i available – the non-exhausted send window *SWND* – for the dispatch of a data segment. Based on the environment or network protoclol used, the *SWND* might be an information provided by a transport layer CC (e.g., MPTCP) or is calculated based on known path capacity and current usage as in the GRE HA.

Algorithm 1: CPF scheduler logic to return the least cost path for each dispatch of a data segment.

Initialization:
    $paths[] \leftarrow [[p_1, C_1], ..., [p_i, C_i]], \quad mincost \leftarrow max$

for $e \in paths$ **do**
    if $(e.C < mincost)$ AND $(SWND(e.p) > 0)$ **then**
        $bestpath \leftarrow e.p$
        $mincost \leftarrow e.C$

return $bestpath$

This generic logic is adjusted for the implementation in MPTCP in Listing 3.3. While in Listing 3.1 the SRTT values are compared to select the socket related to the subflow with the fastest propagation, Listing 3.3 considers a cost metric instead, following the path cost parameter C_i defined in Equation 3.1. This requires a C_i stored along with the network interface, where an evaluated subflow corresponds to, reflects its value in the `prio` variable, with a lower value indicating a lower cost.

In Linux, the tree `/sys/class/net/*` keep diverse information for each network interface device, with `*` being a placeholder for a real device name such as `wlan0` (Wi-Fi) or `wwan0` (Cellular). The Linux Kernel documentation[4] provides an overview about the information which can be collected and the settings which can be executed under the particular device tree by simply reading and/or writing to files. For example `/sys/class/net/*/carrier` holds the information about the physical link state. Facilitating the implementation of the *CPF* scheduler, a new file `mptcp_prio` is added to the network device tree at `/sys/class/net/*/mptcp_prio`. This one is envisaged to carry the network related cost information, which can be finally read out by the *CPF* scheduler through a modified socket link structure with `tp->mptcp->link_info->mptcp_prio`.

The final decision logic in the *CPF* scheduler is the same as above for the *default scheduler*, but considers the `prio` value, instead of the `tp->srtt_us`. For this purpose the content of the `mptcp_prio` file is read out from the network interface a subflow belongs to. Paths with lower `prio` values are preferred over those with higher `prio` values and this is compared against the `min_prio` (*default scheduler*: `min_srtt`) value, which holds the value from the previously evaluated subflow within the loop. Also here, the `min_prio` value has to be initialized with a maximum value for a defined start of the scheduling loop.

Taking the network interface names used above and assign the cost parameters as outlined in Listing 3.2 through a Linux terminal, will force the *CPF* scheduler to squeeze the traffic

[4] https://www.kernel.org/doc/Documentation/ABI/testing/sysfs-class-net

into the Wi-Fi subflow as long as the subflow related send window permits. Otherwise, the traffic is sent via the cellular subflow.

```
echo 0 > /sys/class/net/wlan0/mptcp_prio
echo 1 > /sys/class/net/wwan0/mptcp_prio
```

Listing 3.2 Access cost definition through Linux `/sys/class/net/` tree

To address the case when same costs are given to interfaces and avoid an unpredictable subflow selection, a small enhancement ensures that in this case the subflow with the lowest SRTT is preferred. This means if several paths exists with the same cost, the *default* logic is used to prefer then the path with fastest propagation. In Listing 3.3 this enhannced logic is part of the if-statement after the `OR`.

```
prio = tp->mptcp->link_info->mptcp_prio;

if (prio < min_prio OR (prio == min_prio &&
     tp->srtt_us < min_srtt)) {

   min_srtt = tp->srtt_us;
   min_prio = prio;
   bestsk = sk;
}
```

Listing 3.3 MPTCP CPF path selection logic implementation within scheduler

In the obvious case of a terminal – smartphone or CPE – implementation, the individual access interfaces are terminated in these devices and can be used to store the respective cost indicator. In far-end systems like servers in the Internet or the 3GPP-ATSSS/HA termination point in an operator core, it is likely that a network interface has no direct relationship to the access network used in the terminal. In this case the MPTCP in-band policy exchange can be used to configure and update the cost parameter of a path on one or both sides. For example in a two access path scenario, the `MP_PRIO` option with its "B" Bit, defined as part of [10], can be exploited to distinguish the costs between the path using the first access and the path using the second access. This helps in scenarios

- when there is no 1:1 mapping of access interfaces between the MPTCP termination points.
- when only one side provides an interface for cost updates.

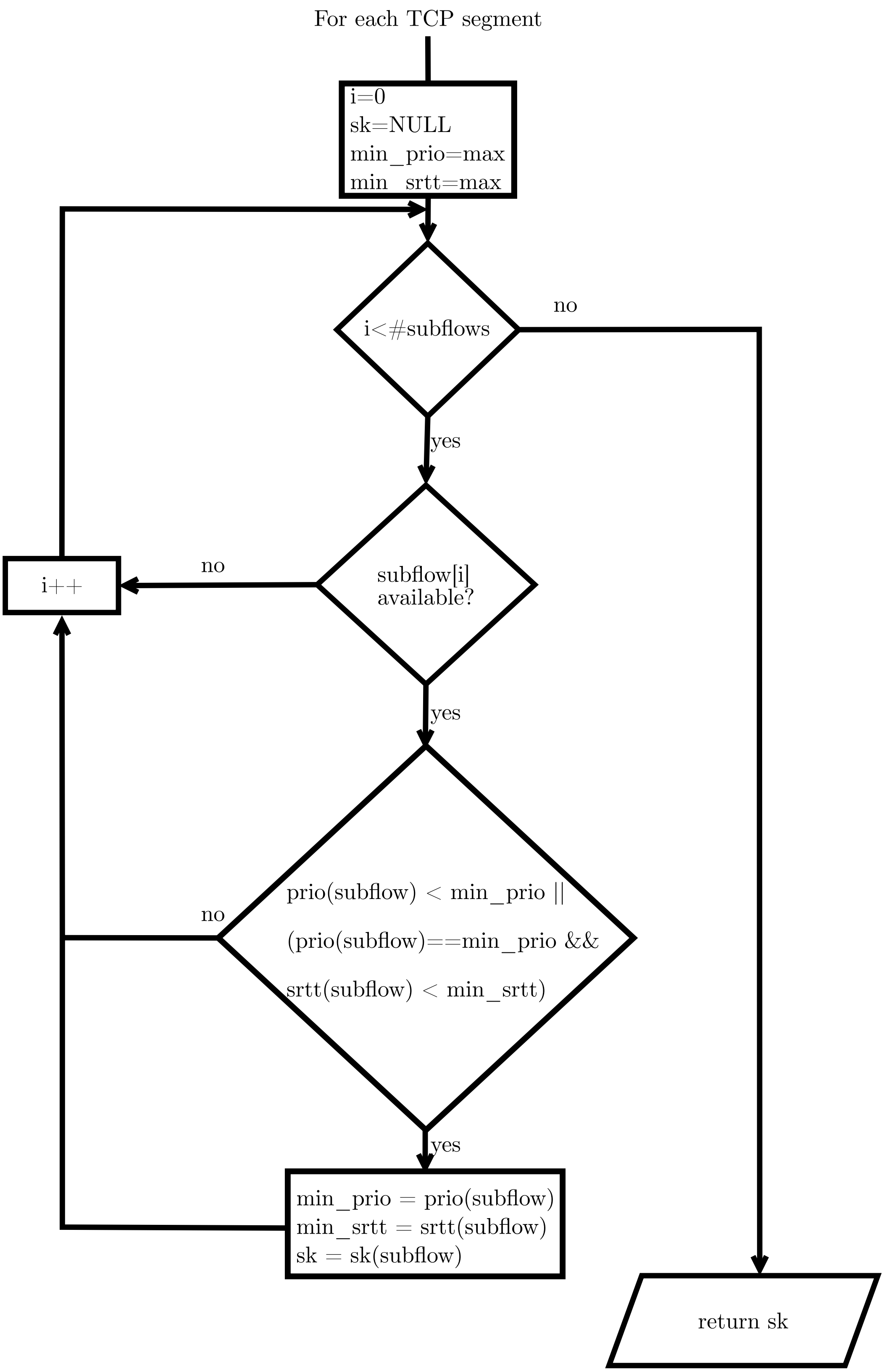


Fig. 3.2 Flow diagram MPTCP CPF scheduler

The entire process of the MPTCP *CPF* scheduler from the availability of a TCP segment to the socket selection for sending is shown in the Figure 3.2 flowchart. Whenever a TCP segment is available to send, the scheduling logic starts to initialize a loop counter `i`, a socket pointer `sk` and two variables `min_prio` and `min_srtt` set to the maximum value they can hold. The first step within the loop is to verify if further iterations over the number of subflows `#subflows` are required. This will for example fail if no subflow is established or available at all or the variable `i`, which is incremented after each iteration, indicates that all subflows have passed through. In this case, the scheduling logic stops and returns no socket in the first case or the `sk` related the subflow with the lowest priority (cost) and if necessary with the lowest `SRTT`. Returning a valid socket requires, therefore, that the loop is run at least once. So at least one `subflow[i]` out of the list of subflows has to be available with a *snd_wnd* $\neq 0$. If this is true and the path priority value `prio(subflow)` is lower than `min_prio`, the current subflow socket `sk(subflow)`, the priority value and the SRTT information are stored in the respective variables `sk`, `min_prio`, `min_srtt` and are available in a next iteration over the next subflow. The stored values are only overwritten when a subflow is available with lower priority than the one stored or on case of same priorities if the SRTT is lower. As described above, the successful determination of a socket `sk` with the most least cost is finally returned.

3.2.4 Verification of the CPF principle

This section presents a review of the functionality of the CPF MPTCP implementation from subsection 3.2.3. For this reason a simple local testbed is used with two PCs, equipped both with two Ethernet interfaces and one Wi-Fi interface supporting the IEEE 802.11g standard, running the MPTCP reference implementation version 0.94, including the new developed *CPF* scheduler and prioritization interface. The detailed testbed description used at later points in this work is available in section 4.1.

To investigate the strict prioritization of subflows in MPTCP, the *CPF* scheduler is compared with the MPTCP *default* scheduler. For this, all three interfaces of the PCs are connected one to one, with the Ethernet interfaces limited to 100 Mbps to produce diagrams with a reasonable scaling, according to the maximal achievable throughput of 54 Mbps in a 802.11g Wi-Fi system. The MPTCP path manager, responsible to establish and maintain the subflows, is forced to open a subflow per interface, resulting in three subflows. As a traffic generator iPerf3[5] comes into use which is capable to send TCP traffic with different

[5] https://iperf.fr/iperf-doc.php

throughput speeds.

For a first insight, the schedulers are compared in situations with fluctuating throughput demand to make the traffic distribution difference visible. The throughput demand is increased in the following interval: 10 Mbps → 50 Mbps → 150 Mbps → ∞.

The objetive of this testing is to get the Wi-Fi related subflow prioritized over the first and the second Ethernet access and to show that the *CPF* scheduler provides predictable results in terms of prioritization while the *default* scheduler does not.

Starting with the *default* scheduler, the first interval in Figure 3.3 shows that both Ethernet paths are randomly used when 10 Mbps is transmitted, even though one Ethernet alone could meet the demand. With increasing speed the random usage stops and one Ethernet link is used solely (50 Mbps) or in combination if the demand exceeds one Ethernet link (150 Mbps). Wi-Fi is only added, if both Ethernet links are not able to deliver enough throughput, which is the case in the unlimited scenario (max). All in all, the traffic split is not really predictable, however the throughput performance is very well and the links are operated at their limit.

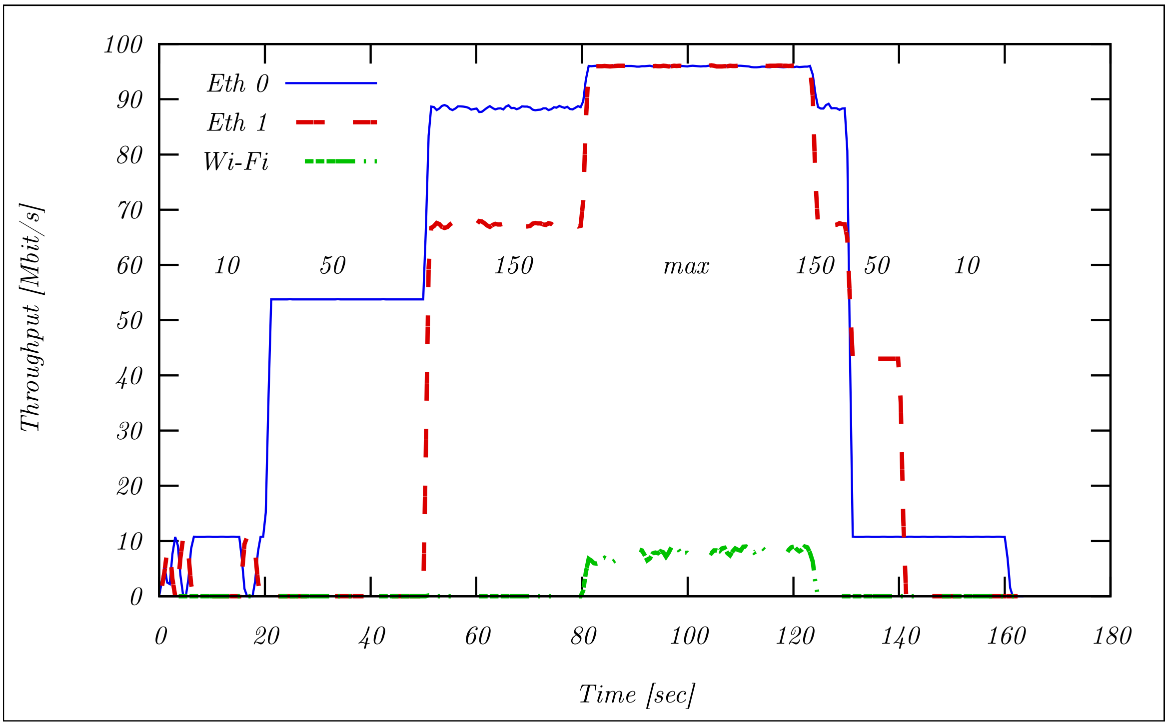


Fig. 3.3 Traffic split MPTCP default scheduler

If *CPF* scheduler is used, a predictable behavior is established as shown in Figure 3.4. The Wi-Fi is prioritized, with a cost value of 0, over the first Ethernet link (*Eth 0*, cost value 1) and this one again over the second Ethernet link (*Eth 1*, cost value 2). The same speed interval is applied as for the previous test. At the beginning, different to the *default* scheduler, the Wi-Fi link is solely in use as the requested 10 Mbps can be provided. In the next step, the generated throughput is increased to 50 Mbps, which leads to a total usage of the Wi-Fi link with 15 Mbps and the remaining 35 Mbps overflows to the next prioritized *Eth0* link. A further increase of the throughpout requires *Eth1* to jump in and an unlimited speed leads to full utilization of all links. While the throughput performance is absolutely the same between both, the *CPF* and the *default* scheduler, a predictable traffic distribution considering access costs is only given with the *CPF* scheduler.

This seems to make the *CPF* scheduler a strong candidate for scenarios which profit from the high throughput capability given in a multi-path system combined with economic steering decisions. Furthermore it confirms that the CPF principle which is is known from HA scenarios can be also implemented in MPTCP.

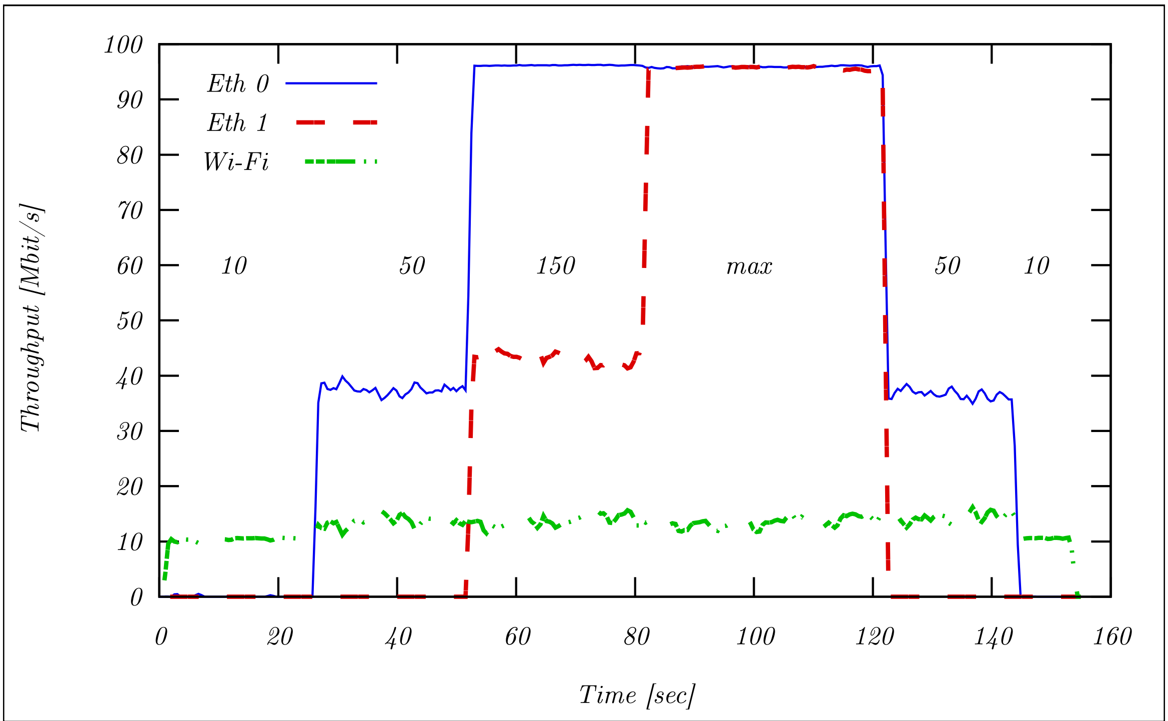


Fig. 3.4 CPF: Prefer Wi-Fi link over a first and a second Ethernet link

After this first demonstration of the *CPF* scheduler capability, still two further functionalities can be tested to get the full picture. The first functionality is the handling of paths with equal costs and the second is the change of costs under operation using the `mptcp_prio` Linux *net* device tree information as shown in Listing 3.2. The setting of same path costs should lead to a *default* scheduler logic for those paths, according to subsection 3.2.3. In Figure 3.5 this is verified for setting same path costs to the Ethernet interface, while the Wi-Fi link has least cost. As expected, the Wi-Fi subflow is used first and during second 20-40, when the throughput exceeds the Wi-Fi capabilities, both Ethernet interfaces jumps in, showing a random usage of both links similar to the behavior of the *default* scheduler in Figure 3.3. Due to the selection of the Ethernet link with the lowest SRTT, after the Wi-Fi path is exhausted, a non-deterministic subflow selection kicks in. The advantage is that the transmission benefits from the best propagation when the cost is not critical.

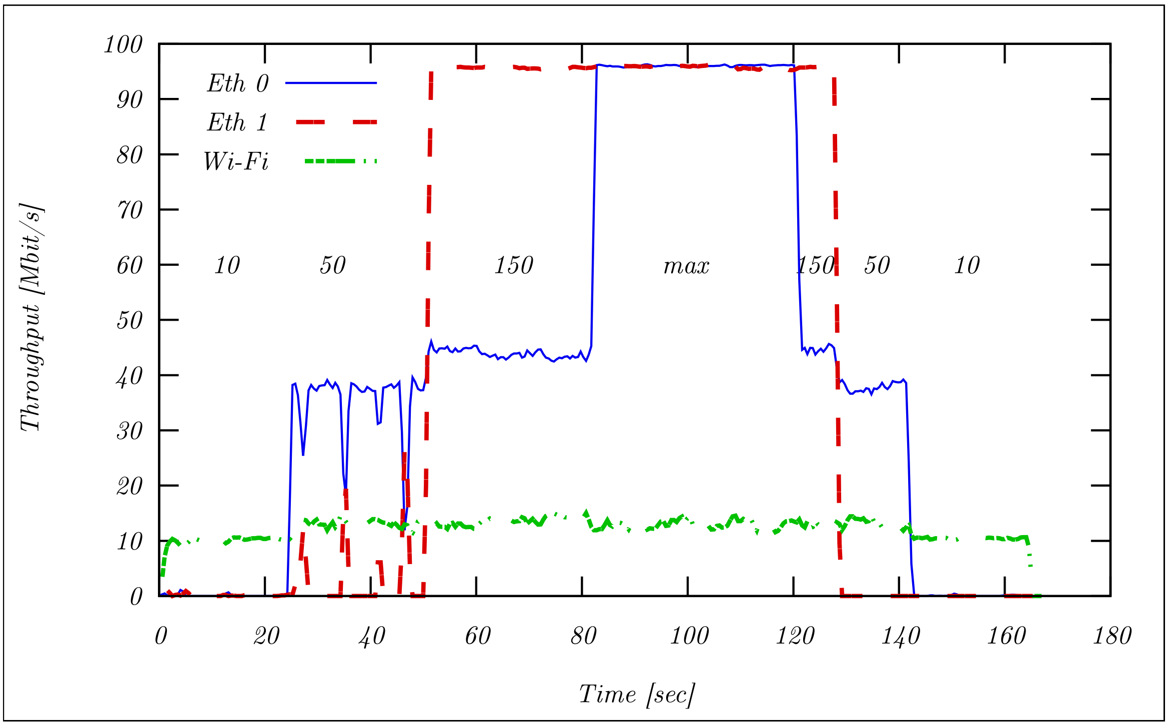


Fig. 3.5 CPF scheduler: Prefer Wi-Fi link over two Ethernet links with same cost

With toggling the path costs under operation the strength of the MPTCP *CPF* implementation is demonstrated in Figure 3.6. In a simplified setup only one Ethernet and Wi-Fi interface are used and the cost values are toggled using the commands from Listing 3.2. As expected, the *CPF* scheduler prioritizes the Wi-Fi or the Ethernet subflow according to the cost values received through the net device tree when the network interface related `mptcp_prio` value is updated. This becomes clearly visible, when the Ethernet interface is prioritized with its capacity higher than the transmitted 50 Mbps, then no Wi-Fi resource is touched. In contrast, the Wi-Fi connection is used to the maximum when prioritized, and the remaining throughput is provided via the Ethernet connection. **This provides tremendous flexibility in systems where costs are expected to change during operation.**

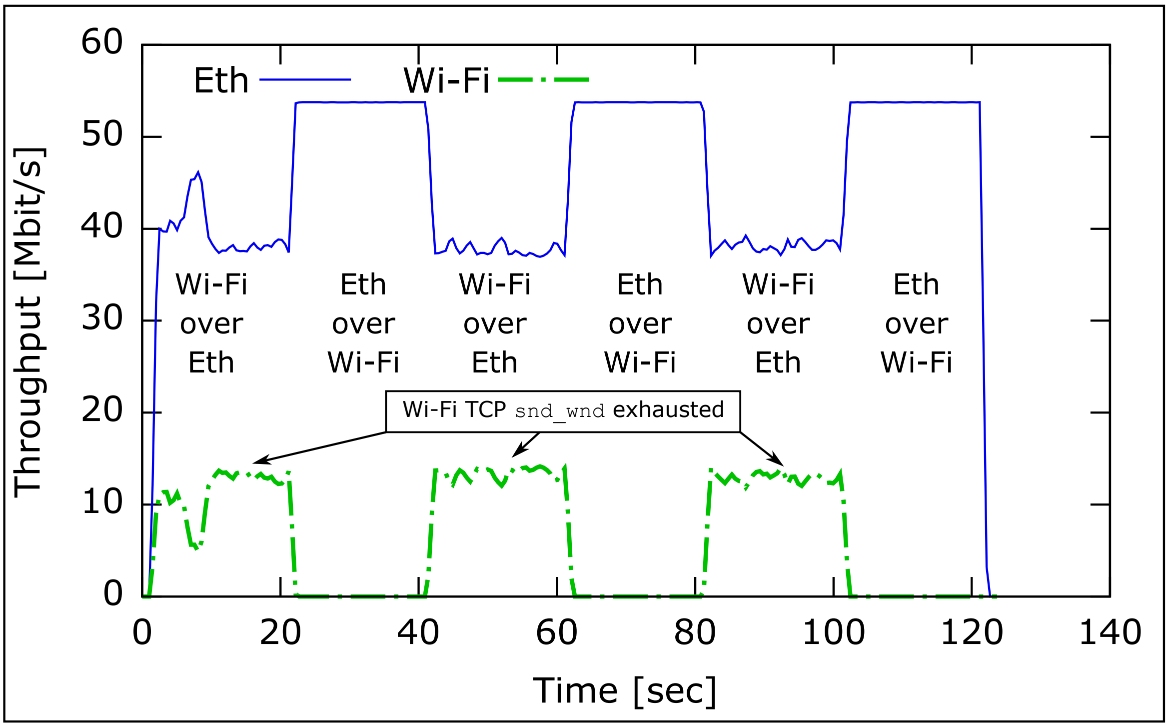
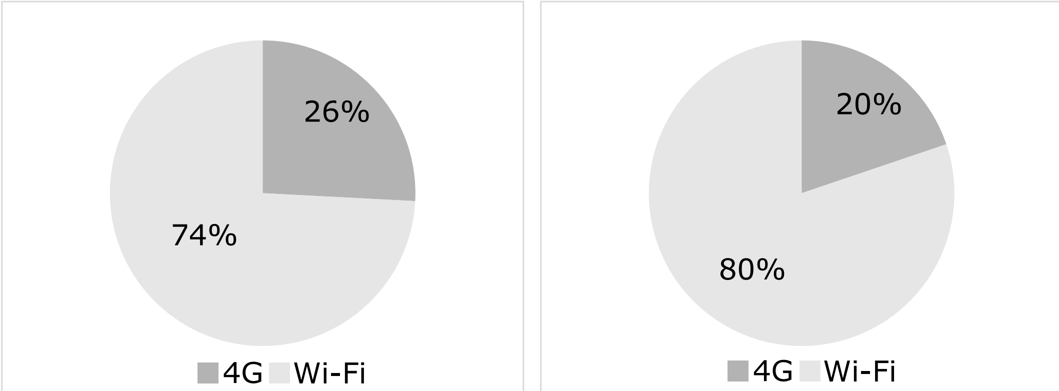


Fig. 3.6 CPF scheduler: Toggle link cost

Another important starting point for the verification of the CPF principle has been the results (Figure 3.7) of a real user trial study, conducted with 5 mobile customers of a MNO, using the online testbed with mobile phones described in subsection 4.1.3 for a 3GPP ATSSS like experience. The objective of this study was to evaluate the effect of *CPF* compared to the *default* MPTCP scheduler, based on lower latency (SRTT). Both schedulers were applied each over one month to every TCP communication transferring each a total of 71 GB (*default*) and 77 GB (*CPF*). Figure 3.7 shows the reduction in the use of the costly LTE access when *CPF* is applied over the SRTT driven *default* one. This confirms the erratic path selection of the *default* scheduler from a cost perspective.



(a) Default (b) CPF

Fig. 3.7 Customer trial - Downlink LTE share comparison *CPF* and *default* scheduler

3.2.5 CPF limitation and Problem statement

According to the first tests in subsection 3.2.4, the CPF principle seems to work fine for constant bitrate traffic as it was generated with the `iPerf` tool. Also the flexibility of the cost scheme applied to the network interfaces was demonstrated. Since typical Internet communication does not consist only of constant bitrate traffic, as it could be find for example for file downloads, verification of the CPF principle using different traffic patterns than constant bitrate is advisable.

The Cisco virtual Networking Index [17] and other studies [18, 19] give a clear indication that VoD services dominate the world-wide traffic-mix with a share estimated at 80 %. It is therefore obvious to confront the MPTCP *CPF* implementation with the typical traffic pattern generated from VoD services which is known to be different from constant bitrate traffic. This is especially the case when the transport capacity is higher then the VoD service bandwidth required for smooth playback. In this scenario, traffic patterns show a bursty character that is more pronounced the higher the service and transportation capacity. The different shapes of possible VoD traffic patterns are depicted in Figure 2.6, while the cause is discussed in section 2.4.

To approach realistic scenarios when CPF is tested with VoD traffic, the simple test environment from the initial *CPF* tests in subsection 3.2.4 was extended to include access to the Internet. A more detailed testbed description is provided in subsection 4.1.2 in the course of describing testbeds for testing the outcome of this work – an enhancement of CPF. In short, compared to the setup for CPF verification with constant bitrate traffic in subsection 3.2.4, a MPTCP termination point is moved to a server on the Internet. For an interworking with VoD service provider, a TCP-Proxy[6] on this MPTCP termination point forces the conversion between the TCP stream established with the VoD provider and the MPTCP connection towards the consuming terminal device. Also the accesse types used in the local terminal changed from former Wi-Fi and Ethernet links in subsection 3.2.4 to 4G and DSL access to resemble HA. A point that is not decisive, but mentioned for the sake of completeness, is that an underlying User Datagram Protocol (UDP) tunneling was applied to connect the local MPTCP terminal with the one in the Internet. That has the benefit, that first, the transparent Proxy from footnote 6 can be used without explicit addressing using Linux TPROXY[7] functionality. As a second benefit, the MPTCP header options are hidden from middle-boxes which might prevent MPTCP from working when MPTCP required TCP header options are filtered or blocked, as outlined in [10, sec. 6]. For this purpose, a Linux Kernel implemented tunneling software[8] is used in a modified version, which implements UDP tunneling.

The objective of the following test is to understand whether the phenomenon of spurious cost increase, which is relevant to the GRE HA (section 1.1) is a general CPF problem. For this purpose, a VoD transmission over a 6 Mbps cheap DSL path is compared with and without access to the LTE path. While, as shown in Figure 3.8, the DSL single path scenario the requested Full HD (1920x1080) video[9] was transmitted without any negative impact on the experience - smooth playback - combining with LTE using MPTCP CPF implementation results in a significant load on cellular resources.

Again, it is important to note that in both scenarios the video runs smoothly. The fact that the *CPF* scheduler transmits around 90 % of the traffic over the costly LTE path, despite the fact that the LTE path should be of lower priority, presents a major limitation in terms of the application of this scheduler.

This behaviour of the *CPF* scheduler has its origin in the nature of the VoD traffic. The VoD typically exhibits a bursty traffic pattern (Figure 1.2, Figure 2.6), using transport protocols like TCP or QUIC. Such protocols grab as much throughput as they can, which is also in the interest of the VoD operator, giving better overall service to the users and avoiding playback issues. On the receiver side, the playback application maintains a buffer storing the portions of the video until they are played. The new portions of the video are only transmitted

[6] https://github.com/VRT-onderzoek-en-innovatie/tcp-intercept
[7] https://docs.kernel.org/networking/tproxy.html
[8] https://github.com/telekom/tunprox
[9] https://hls-js.netlify.app/demo/?src=https%3A%2F%2Ftest-streams.mux.dev%2Fx36xhzz%2Furl_8%2F193039199_mp4_h264_aac_fhd_7.m3u8

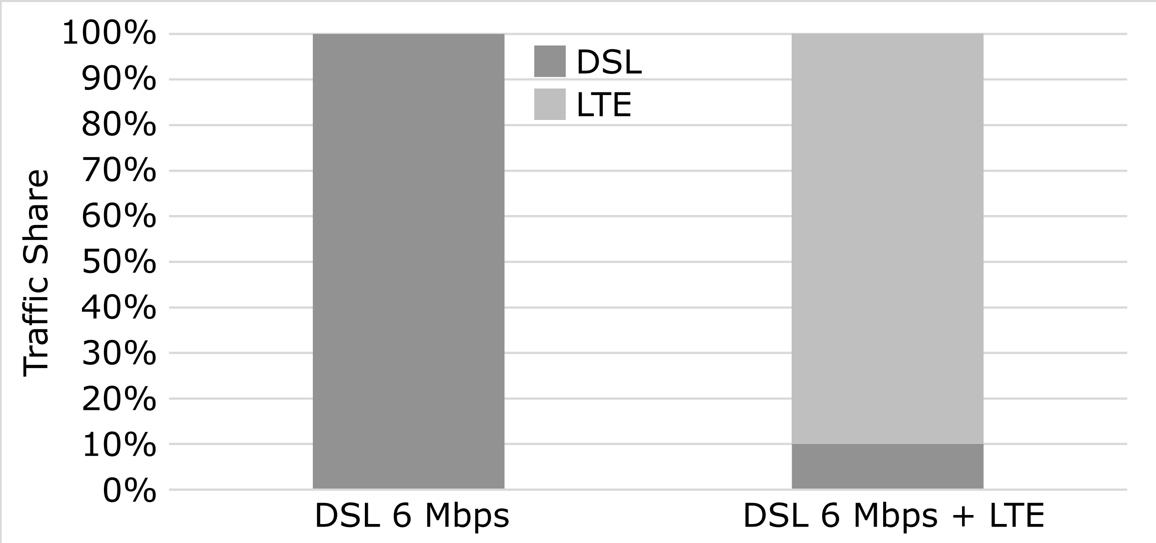


Fig. 3.8 Traffic share for smooth HTTP VoD streaming with 1920x1080 H.264 over 10min comparing single path DSL and multipath DSL + LTE with MPTCP and *CPF* scheduler

if the buffer level has reached a low state. That avoids unnecessary transmission of the whole video while ensuring smooth playback, albeit it is responsible for the bursty traffic pattern as long as the bottleneck throughput is sufficient to transmit portions of the video faster than being played. Two issues appear to challenge the use of the *CPF* scheduler in this context. The first is the very fast dispatch of portions of the video data from the VoD service. This is most often the case at the proxy, also denoted as HA Gateway or ATSSS UPF, with significantly higher throughput then the bottleneck-throughput towards the consumer. The second issue is the time until a proper `snd_wnd` is built up on the primary path. Both of these issues lead even at higher DSL datarates with 50 Mbps or 100 Mbps to an increased LTE share, measured at ~40 % and ~25 % respectively. One solution to this problem may be to increase the capacity of the cheaper access network using fiber or other technologies. The existing Data plan limitations, wide deployment of copper based infrastructures and Wi-Fi bottlenecks however often lead to situations where the capacity of the primary path is limited.

Overall, the study conducted here, with a *CPF* scheduler implemented in MPTCP, confirms the different indications given from a HA operator in section 1.1 and the authors of [92] for a smartphone scenario. All presented and discussed a conflict between providing high throughput capabilities using multi-path and the onloading of traffic on expensive path resources, even if *CPF* is applied, for mobile and stationary scenarios.

It is therefore obvious, that multi-path setups which implement cost efficient steering, as defined in BBF and 3GPP for HA and 5G ATSSS, suffer from spurious demand on the non-preferred path, when it comes to the transmission of bursty traffic as it is likely to appear when VoD is consumed. With the increasing share of VoD consumption and the continuation of multi-path deployments, this presents a very important research challenge.

3.3 Summary of the problem and research objective

When multipath networks for path aggregation corresponding to the model in section 3.1 meet the requirement of cost-efficient multipath traffic distribution, this initially sounds like a solvable task. Under the prerequisite, that the individual path costs C_i and provided path throughputs B_i (or the remaining path throughputs) in a multi-path system are known, it simply requires to prioritize the path with the lowest cost for traffic transmission. A path with a higher cost will only be used in addition, if the traffic demand exceeds the capacity of the low cost path. This principle is implemented by the *CPF* scheduler described in subsection 3.2.3 and leads to the expected result that traffic is always routed to the path with the lowest cost that has free transmission capacity. This can be visualized very well when constant bitrate traffic is transmitted below the multipath sum throughput B_{mc} as defined in Equation 3.3. The expensive path resources are used only in the amount corresponding to the difference between the total available bandwidth B_{mc} and the low cost path bandwidth.

As of 3GPP Rel. 16, this principle is also defined for 5G systems [11, sec. 4.2.10], known there as ATSSS, providing simultaneously access over cellular and non-cellular connectivity in mobile and adopted by BBF for residential [12] scenarios. With the *Priority-based* scheduling logic as part of the ATSSS rules specified *Steering Modes* in [11, sec. 5.32.8], the CPF principle is incarnated. Its implementation can be expected as part of the ATSSS higher-layer (ATSSS-HL) functionality using MPTCP [10], as this one provides the information about the path utilization, which is essential in the scheduling process to recognize when a (low-cost) path is saturated and a second (expensive) path has to step in. The ATSSS lower layer (ATSSS-LL) functionality, providing multi-path support for Link and Internet layer, is lacking such information as no measure is defined yet to gain those under operation.

A *CPF* scheduler implemented in MPTCP demonstrates in subsection 3.2.4 the flawless operation for a scenario with a constant throughput demand, but on the other hand emphasize the tension between multi-path costly resource usage in a VoD playback scenario without a QoE benefit in subsection 3.2.5. Similar behavior was monitored in HA environments as introduced in section 1.1.

The bursty nature of the VoD traffic as outlined in section 2.4 challenges the concept of cost efficient multi-path transmission as it operates often with high peak datarates and therefore make use of the aggregated throughput, up to B_{mc}. On the other hand, section 2.5 teaches that VoD QoE is mainly impacted by stalling events and low video resolutions which ultimately has a dependency on a certain transmission throughput, but not necessarily on peak datarates with which the burst were generated. Along with the fact, that the VoD traffic was forecasted to be the dominating component of the Internet traffic, with more then 80 % share stated in [17], motivates the development of new more efficient cost-based scheduling solutions.

[92] confirms this conclusion with the same observation of an extraordinary usage of costly path resources for VoD transmission over multi-path systems. The proposed MP-DASH (see also section 2.1), however, requires a strong interaction between DASH client and a stateful multi-path network protocol such as MPTCP, for which the specified *Deadline-Aware* scheduling algorithm reduces impressively the costly resource usage as long as QoE impairment can be excluded. This limitation disqualifies itself for any non-DASH based VoD transmission and more important for multi-path transmission systems which are not able to interact with a playback client like the proxy architectures HA and ATSSS.

It is therefore an objective of this work to overcome above limitations and find a multi-path scheduling algorithm which is agnostic to VoD service providers, VoD transmission protocols and multi-path network protocols and which minimizes the spurious costly resource footprint when it comes to VoD transmission without impairing QoE. This is subject to two-dimensional optimization problem for which the multi-path scheduler has to provide at any dispatch decision point a path resource which facilitates the best possible QoE according to the characteristics of the composite multipath link mc (Equation 3.2) and on the other hand keeps the resource overhead C_{mc} (Equation 3.4) at the bare minimum.

Developing and investigating algorithms for this optimization problem, with its dependency on QoE and cost, is the main task within this work.

Although a solution at this point is considered independent of the deployment architecture of the multipath system, end-to-end or **transparent** proxied (HA, ATSSS), the latter raises special demands which needs to be considered in a solution:

- Integration with Internet services like Youtube is not an option due to the diversity of existing services
- Traffic inspection is inhibited tue to proliferation of encrypted traffic
- Services like a speedtest or an unlimited file download must not be affected
- Efficient implementation to handle the high expected throughput

3.4 Simplified QoE determination and dependency

It is imperative to look again into the determination of QoE after section 3.3 outlined the need for a scheduling algorithm beyond CPF which optimizes cost in the presence of bursty VoD traffic without loosing the performance of multipath transport to gain better QoE.

With the definition of MOS as elaborated in section 2.5 a tool is available to determine QoE in a quite laborious procedure. However, due to own experience VoD transmission most often suffers from limited throughput capabilities which lead either to playback interruptions or a selection of a lower video resolution. Starting from this angle analyzing the input parameters for the MOS calculation, three major parameters crystallized:

- Time $t_{initial}$
- Time t_{stall}
- Video resolution r

$t_{initial}$ is the time until a video starts after it was requested. t_{stall} is the amount of time the video stopped playing during playback. r is the played out video resolution. According to [124] these parameters have the highest impact as long as the following prerequisites are considered across all test scenarios:

- same video codec is applied
- Video compression degradation follows r degradation
- The frame rate is static

- Audio quality follows the video quality
- The device under test supports the maximum display resolution of 1920x1080 pixel
- Equal measurement duration

This has the benefit that a laborious MOS calculation can be given up in favor of a quantitative comparison of these three parameters.

$t_{initial}$, t_{stall}, and r determined in a multipath system as defined in section 3.1, are typically dependent on the time-varying characteristics of the transmission path according to Equation 3.1 when only one path is used, or according to the aggregated path characteristics Equation 3.2 when multiple paths are used.

In the special case of non-realtime Internet services where VoD belongs to, measures are implemented to compensate Internet typical path characterstics related to the time it needs to get data successfully exchanged between endpoints. In the case of VoD, for example, this is the client receive buffer $RecvBuffer_{VoD}$ (section 2.4), which is of a size that eliminates latency, jitter, and loss rate.

This means, that the QoE of VoD is mainly impacted by the available throughput and other path capabilities play a negligible role. This is because the receiver buffer compensates, as long as the BDP is covered by the $RecvBuffer_{VoD}$ and paths p_i exceeding this size with $B_{mc} \cdot 2(L_i + |J_i|) > RecvBuffer_{VoD}$ or with re-transmission included $B_{mc} \cdot 4(L_i + |J_i|) > RecvBuffer_{VoD}$ are removed from the multipath system. Under this condition QoE path characteristic dependency can be reduced to throughput B_p with a VoD QoE definition of:

$$QoE_{VoD} = \mathfrak{F}(B_p) \tag{3.6}$$

3.5 COM algorithm description

In the course of developing a solution to overcome the limitations of the *Cheapest-path-first (CPF)* scheduler, a new scheduler called *Cost Optimized Multi-path (COM)* is developed. The idea starts with a simple finding in subsection 3.5.1 and becomes tangible in a first theoretical design in subsection 3.5.3, which follows established design principles from subsection 3.5.2. Early tests and considerations required a refinement of the *COM* design in subsection 3.5.4 to cope with real traffic requirements from Internet services. Finally, the

MPTCP Linux Kernel *CPF* scheduler developed (subsection 3.2.3) is modified to incorporate the COM design in section 3.6 which turns out to be a straightforward extension.

3.5.1 Basic idea

Having now the basic structure and understanding of the critical features and challenges of VoD delivery over cost efficient multi-path systems, let's start to develop an idea how an optimized and service agnostic handling of VoD traffic under cost aspects in multi-path scenarios can be implemented. Before jumping into tangible algorithm development, a reflection of the important outcomes so far will help shaping optimized scheduling mechanisms.

1. Depending on the amount of bandwidth of the transmission path B_p, a VoD service forms a characteristic burst traffic pattern tp_{VoD}, see section 2.4.
2. VoD QoE is depending on various input and output parameters and relies on a proper transport, see section 2.5.
3. Multi-path transport performance is depending on various path characteristics with non-linear dependencies, except total throughput section 3.1.
4. The cost of single-path transport is increased when a multi-path system aggregates paths with higher costs, see section 3.1.
5. Simple path prioritization for cost optimization in multi-path networks is unable to perform for VoD traffic patterns see subsection 3.2.4 and section 1.1.
6. With MP-DASH, a service dependent solution for VoD multi-path cost optimization using DASH is proposed, see section 2.1. However, the strong interaction of DASH client and multipath network protocol disqualifies it for use in the scenarios considered here, which do not allow this relationship.
7. Optimizing **QoE and** multi-path transmission **cost** for VoD playback is a two-dimensional problem as identified in section 3.3.
8. VoD QoE dependency can be narrowed down to the available transmission throughput, see section 3.4.

Using the findings in 1 to 7 make it look difficult to find an optimal scheduling algorithm, considering QoE and transmission cost. The complexity of QoE determination, the non-linear transport characteristics of a composite multi-path link and VoD service agnosticism confronts one with various unknowns to solve this on a mathematical level. On the other hand the reduction of complexity becomes significant in item 8 of the list above, when QoE becomes a function of throughput only (compare Equation 3.6). This is a ray of hope, as it let conclude that QoE is correlated with the available transmission throughput, as long as the composite multi-path link does not harm the allowed Bandwidth Delay Product (BDP) given by the VoD client receive buffer. Since such receive buffers typically hold still a couple of seconds of a video when the buffer falls below a threshold and new video frames are requested from the VoD server, the typical path propagation in terms of latency or RTT, of accesses like DSL, Fibre, DOCSIS, Wi-Fi or cellular 4G/5G, is covered. This let state, that VoD is within some limits delay tolerant.

If this conclusion is drawn, one can think again about what the main problem is. The bursty nature of VoD transmissions uses regularly the aggregated path resources in a multi-path system. This is demonstrated in Figure 3.9, where such a traffic pattern is sketched over time using the CPF principle with a baseline cheap path and an on top stacked expensive path. A detailed analysis of how this traffic pattern arises in section 2.4 shows that an available transmission path bandwidth B_p higher than the required B_{demand} of the VoD service leads to this effect, which is also characterized by a time gap T_{GAP} between transmission bursts.

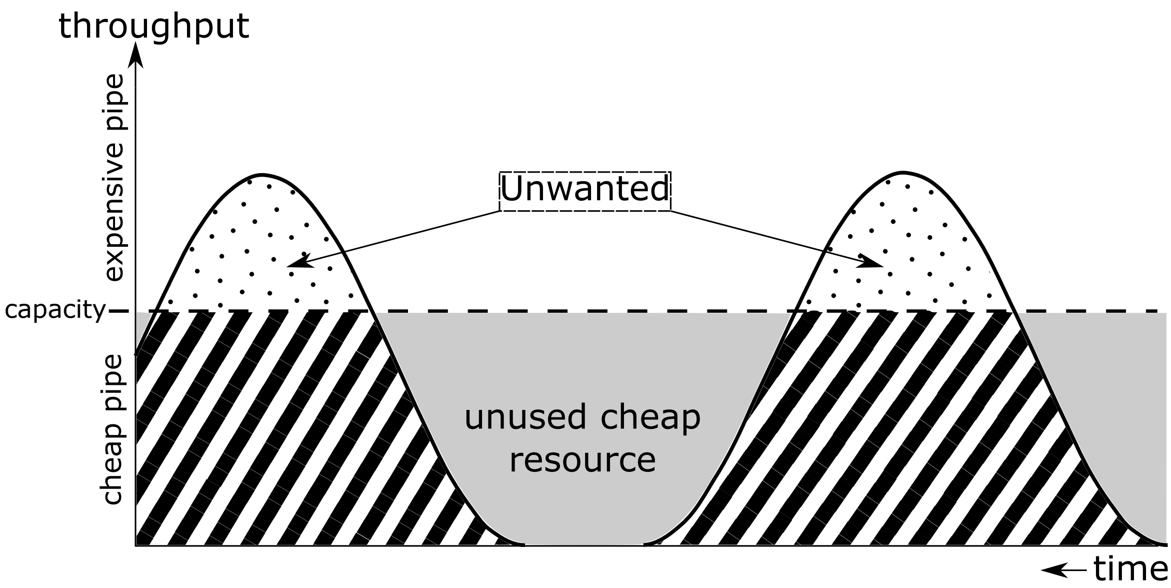


Fig. 3.9 Unwanted multipath operation when traffic burst overflow into costly paths

While the dashed part of the burst is in terms of a cost optimized transmission wanted, the dotted parts are **unwanted**. While this is obvious, it is conspicuous that between the bursts unused cheap path resources are available. From there to the idea of a COM scheduling approach it is quite short and finally presented in Figure 3.10.

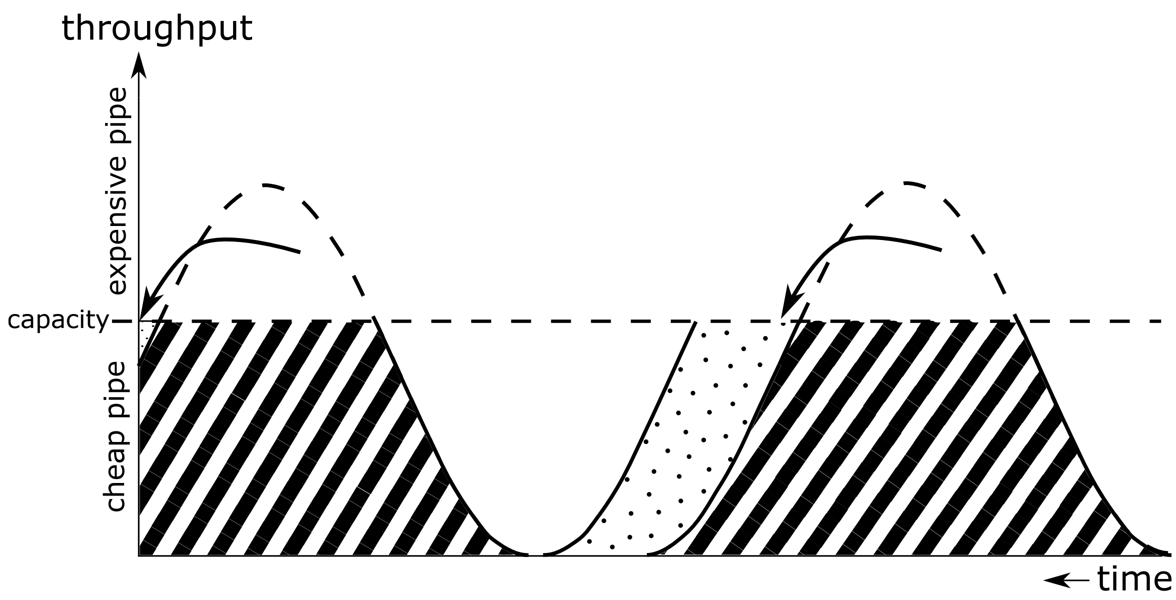


Fig. 3.10 Idea – Squeeze overflowing traffic into the "valley" between bursts

The basic question here is whether it is possible to tilt the peak of the burst into the unused area of the cheaper path if the peak of the burst is overflowing into the expensive area. Instead of a theoretic solution to find an optimum, this is a very practical approach and is further brought to the point in Equation 3.7. If within a time interval $[0,t]$ the throughput provided by the cheaper path and denoted as B_{cheap} is greater than the throughput demand B_{demand} in the same interval, the expensive path resources are not required. Conversely, this means access to the aggregated path conglomerate can be given if the throughput demand cannot be satisfied by the cheap path resource and the CPF principle applies.

$$\left.\begin{array}{ll} only\ cheaper\ path & \text{if } true \\ aggregate\ expensive\ path & \text{if } false \end{array}\right\} \Rightarrow \int_0^t B_{cheap} \geq \int_0^t B_{demand} \tag{3.7}$$

As simple this reads, it still provides a number of pitfalls. This starts with the determination of B_{cheap}, which is, unless it is a static multi-path scenario like HA with fixed access defined as cheaper path and a bottleneck throughput known from the modem synchronization, difficult to detect. As long as the bottleneck is not available in such a predictable manner because B_{cheap} becomes dynamic due to cross-traffic, or the bottleneck is outside the last mile or the cost efficient path is subject to volatile transmission as in the ATSSS scenario with two radio links Wi-Fi and cellular, the bottleneck throughput is subject to estimation and can be solved be selecting a network protocol like TCP or QUIC. Both protocols provide at least the information about the remaining throughput as part of their congestion control procedure. Alternatively, packet dispersion techniques can be used to estimate bottleneck throughput [129].

An additional problem is the appropriate determination of the time interval to probe available and demanded throughput. As long as this is not infinitesimal, there is the risk that the resulting traffic pattern over- or undershoots, meaning the cheaper path is over- or underutilized, which either conflicts with the goal of an optimized QoE or with the goal to minimize cost. Last but not least the demanded throughput B_{demand} as such is not simple to determine, when the design goal is to create a generic and agnostic scheduling algorithm. By claim this excludes a tight coupling with information from the VoD service, it excludes deep packet inspection to gain information about buffer states, requested resolutions, etc. and it also excludes to use of stochastic or heuristic methods. Thus, finding a scheduling solution for optimized QoE and cost under these conditions is difficult and at least subject to inaccuracies, even if B_{cheap}, B_{demand}, and a suitable time interval $[0,t]$ can somehow be determined.

Even if the first idea of squeezing traffic into unused capacity "valleys" of the cheaper path seems to hold pitfalls it could be used as a starting point to develop further ideas and is therefore denoted as the ***COM* principle**. It's now part of the next steps within this work to figure out if an algorithm can be designed which is able to better utilize the cheaper path resources, enhances the CPF design and enables the usage of expensive path resources only if a better QoE can be gained.

3.5.2 Design principles

A principal concern of a COM algorithm should be to avoid spurious demand on costly paths. In VoD scenarios that happens when the access to the costly path in a multi-path system is not regulated. On the other side this approach seems to provide the best QoE when the cheap path alone does not meet the QoE requirements. It is therefore mandatory to actively detect those spurious demands to take countermeasures. This provides two challenges on the detection and the countermeasures. An active detection of spurious demands could be solved by identifying VoD services and their characteristics using Internet Protocol (IP) address information and measurements. Integrated in a service detection unit, this could be used as input for the multi-path scheduling process. Another possibility could be the tight coupling with the VoD service itself, providing a direct interface between service and scheduler. While this seems to be feasible in end-to-end scenarios, the sheer quantity of VoD services and their dynamic address concepts would lead in network operator terminated multi-path scenarios, however, to an increased management complexity. A full coverage of services is hard to imagine in such scenarios and therefore multi-access network architectures like HA and ATSSS does not support this and explicitly emphasize their service transparency. On the *reactio* side it has to be ensured that an elongated scheduling over the cheaper path does not lower the QoE because the service adapts disadvantageously to the cheaper path characteristic (unfavorable video resolution selection) or the cheaper path is overstrained (packet loss). Similar to the reduced complexity due to service agnosticism no limitations in that sense should be introduced due to a dependency on a particular multi-path protocol. A negative example in this context is the MP-DASH [92], which has a strong dependency on the protocol and the service. As more simple and generic an algorithm is, as more likely is its implementation.

Optimizing cost and avoiding complexity and QoE reduction should be the guideline for a COM algorithm design.

With this in mind, one can define design goals for the new algorithm summarized as:

- the algorithm needs to be able to detect traffic bursts causing spurious demand in multi-connectivity scenarios
- the traffic needs to be scheduled as much as possible on the cheaper resources, according to the real application and customer needs.
- the algorithm needs to be generic, simple, and service-agnostic, requiring no service-specific support
- the algorithm must not decrease the user QoE compared to single path transport but should result in a QoE similar to CPF.

3.5.3 First algorithm and Limitations

Using the integral in Equation 3.7 as a decision anchor to allow or deny access to an expensive path, is an outcome of the principal idea to squeeze traffic into the unused capacity slots of the cheaper path (Figure 3.10). Solving therefore Equation 3.7, with the pitfalls of unknown variables – dimensioning the time interval, determining B_{cheap} and monitoring B_{demand} – as described in subsection 3.5.1, plus considering the design guidelines from subsection 3.5.2 is a mandatory but challenging task.

Therefore, starting from a different perspective, a much simpler method is presented in Figure 3.11, without losing sight of the actual goal, and can be used to design a working solution.

Instead of monitoring capacities and demands, the T_{GAP} size can be used to identify the saturation information. To do this, the time gap between consecutive packets can be measured and compared against a threshold value $T_{GAPthresh}$. If the condition given in Equation 3.8 is true, access to the expensive path is prevented for a time span of T_{Delay}, with the multipath scheduler sending the packets to the cheaper path only. As a consequence of this, bursts will be stretched over time using a higher share of the cheaper pipe. Following this idea also means, that the CPF principle stays as it is and is simply extended by a routine which derive from the "valley" size between bursts if the expensive path is available for scheduling.

$$T_{GAP} \geq T_{GAPthresh} \tag{3.8}$$

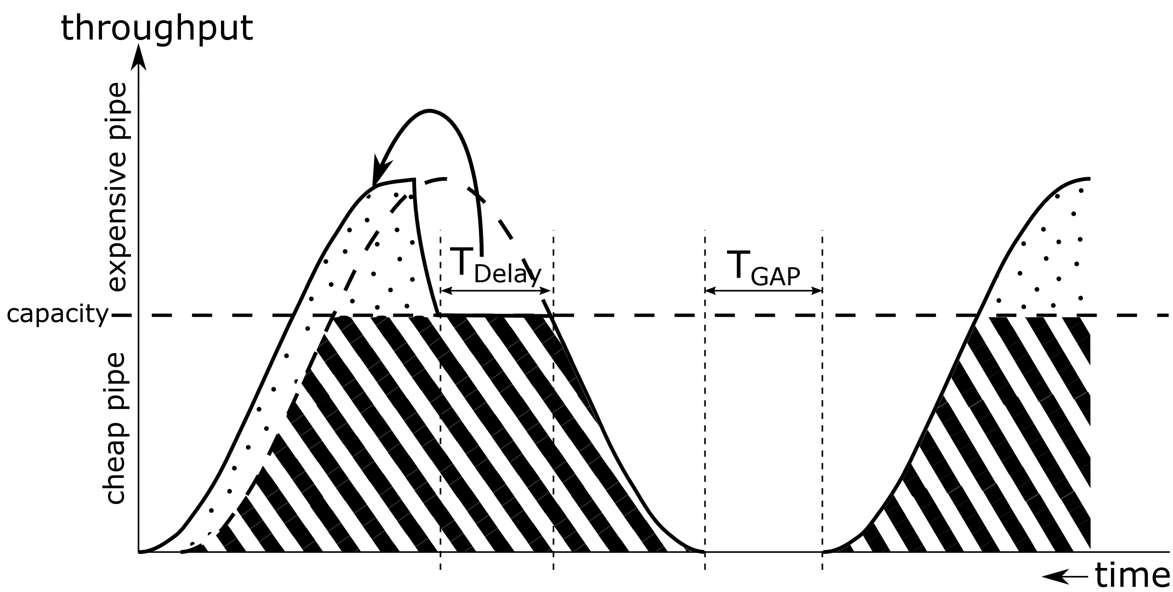


Fig. 3.11 COM - Practical idea to detect unsaturated link capacity based on the gap time T_{GAP} and T_{Delay} as measure to prevent spurious costly demand

The simplicity is captivating. Everything focus on measuring the gap size T_{GAP} and lock, depending on a threshold value $T_{GAPthresh}$ the expensive path for a period of T_{Delay}. The dimension of T_{Delay} finally controls the cost saving. Compared to the integral from Equation 3.7, no time interval $[0,t]$, B_{demand} nor B_{cheap} is required to determine, although the same outcome is achieved. Even if the proposed T_{GAP} detection does not need B_{cheap}, it should be clear, that the general multi-path scheduling algorithm, here CPF, needs this information in order to avoid overloading of the prioritized path.

Matching this basic COM algorithm against the boundaries given by the design principles it can be stated, that traffic burst can be detected, while the self-regulating T_{GAP} measurement keeps the tradeoff between QoE and cost. Whenever the service demand is high, traffic burstiness will start to vanish and the access lock on the expensive path is lifted. This will help to keep access to the aggregated throughput in case of file downloads or when the VoD service tries to iteratively adapt video resolution to higher throughput. The positive side-effect: COM can be applied in scenarios where *CPF* works today, as it fallback to *CPF* scheduling and only optimizes in scenarios where the cheaper path is unsaturated. A more detailed discussion about how COM reacts under different conditions can be found in section 3.7.

The simple key requirement on the multi-path protocol where COM is implemented is the recording of the send timestamps of two consecutive packets, build the difference and compare it with a threshold value in order to decide about locking or freeing access to an expensive path. Since this requirement seems not to be limiting, the COM algorithm can be denoted generic with the potential to be implemented in any multi-path protocol. The computational effort also seems manageable, especially compared to CPF, where COM only has to take additional timestamps, compare, set T_{Delay} and verify. Some of these steps are already implemented in typical protocols, e.g. as part of TCP CC, and can be re-used. In terms of generality it can be also seen, that the number of paths and costs are unlimited. The COM principle, with measuring the burst gap size, applies always to the path where traffic is currently scheduled to. In case a first low-cost path is saturated, COM will give access to a secondary mid-cost path. Providing a third high-cost path will be touched, if a saturation on the mid-cost path is measured, otherwise not.

Despite the euphoria over a seemingly simple approach, it should not be forgotten that with $T_{GAPthresh}$ an T_{Delay} two other unknowns are introduced compared to Equation 3.7.

It is therefore subject to further evaluation studies throughout this work, to verify the assumptions made here. Especially the impact on QoE and cost impact needs to be classified according to a proper selection of $T_{GAPthresh}$ and T_{Delay}.

3.5.4 Final algorithm

The first idea of COM developed in the previous subsection 3.5.3 is captivating in its simplicity and its plausibility cannot be denied. However, evidence needs to be collected to determine if the concept holds and how $T_{GAPthresh}$ and T_{Delay} can be used to regulate expensive path use without compromising QoE. Before discussing the implementation in the next section 3.6, this section analyzes whether other means facilitate practical use.

The COM gap detection process measures the distance between two consecutive packets. When the distance is large enough ($> T_{GAPthresh}$), optimization kicks in and removes the expensive path from the scheduling process for time T_{Delay}. This is considered a valid procedure, but requires that only the VoD material used for the playback process is transmitted. As soon as other data, e.g. signaling information, is exchanged during the bursts, the gap calculation is reset. In the worst case, a single packet between bursts prevents optimization because $T_{GAPthresh}$ is never reached. A measurement of a Youtube VoD transmission confirmed that the time between the transmission of video chunks is arbitrarily used to exchange some data between the video client and the video service. Presumably this is information that helps the service to adjust the playback of video material, e.g., execute Adaptive Bit Rate (ABR).

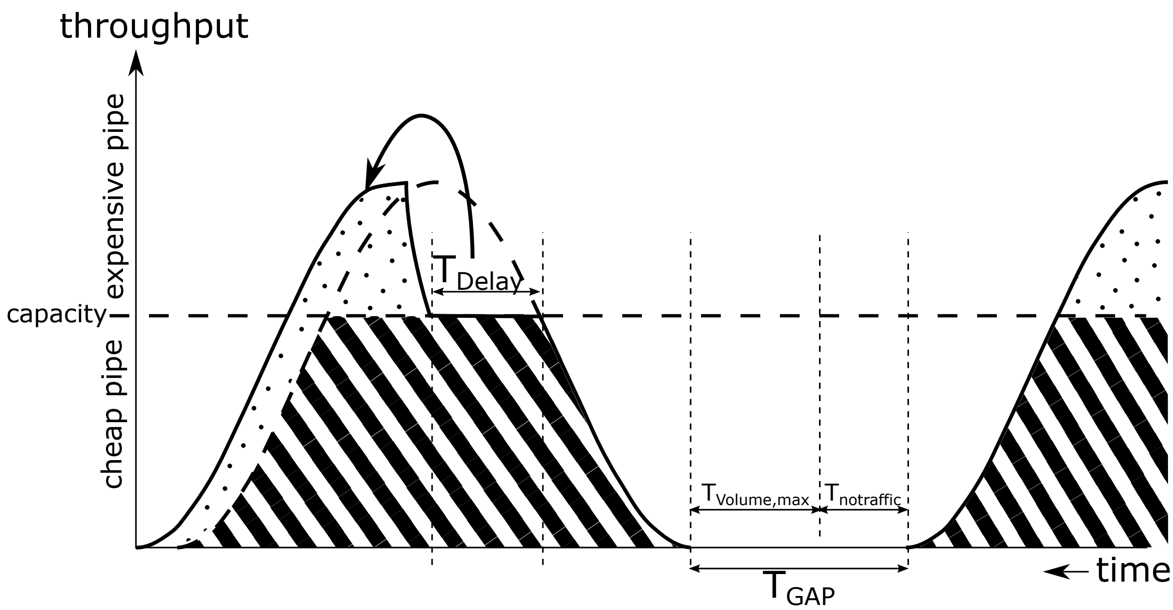


Fig. 3.12 COM - Robust gap detection with $T_{notraffic}$ and $T_{Volume,max}$ specifying periods of allowed or disallowed traffic within T_{GAP} measurement

A measure to overcome this minor traffic, which yields in measuring a small T_{GAP} is proposed in Figure 3.12. Therefore, $T_{GAPthreshold}$ is divided into two subphases $T_{notraffic}$ and $T_{Volume,max}$. If a T_{GAP} size is measured which falls into the time $T_{notraffic}$, the gap calculation resets. So far this is not different from the procedure described in subsection 3.5.3, but if the time $T_{Volume,max}$ is defined (additionally), the gap calculation will not be reset as long as the traffic volume is below a new volume threshold V_{max} with which a certain number of packets or bytes are allowed. But that also means that there are at least two new variables $T_{Volume,max}$

and V_{max} which need attention in the evaluation process. On the other hand, $T_{GAPthreshold}$ is replaced by the sum of $T_{notraffic}$ and $T_{Volume,max}$, which limits the number of new variables to V_{max} unless $T_{notraffic}$ is configured.

Another obvious effect of the algorithm discussed so far is that it works only after a settling phase. This settling phase needs at least one first exchange of data after the connection establishment with a gap larger than $T_{GAPthreshold}$ to start the cost optimization. This excludes services which produce an initial burst right after connection establishment. In this category a website request falls, but also the first transmission of a VoD chunk. Especially the latter is important to consider, because the first burst is often the largest to initially fill the buffer.

Applying a proactively suspension of the high-cost path using $T_{StartDelay}$ as depicted in Figure 3.13 directly active after connection establishment, will avoid a cost intensive early overflow scenario. While this helps to contain the initial VoD burst, it also has an impact on website requests, especially when communicating with HTTP from version 1.1 onwards, where multiple requests and responses are combined into one pipeline, leading to media-intensive websites requiring a lot of transmission resources. However, without knowing whether the addition of an initial access delay is a risk or a benefit, it is worth examining the impact in typical scenarios within the scope of this work. Therefore, it is recommended to review scenarios that generate only a single burst.

The individual measures of fine granular gap detection and optimization of the initial behavior, can also be combined as depicted in Figure 3.14. This is the target image for which implementation is being considered and thus provides the most opportunity to study the performance and impact of each parameter:

- $T_{Volume,max}$: A time during which consecutive packets are not considered in the gap calculation as long as they are within the bounds of V_{max}. Can be combined with $T_{notraffic}$
- V_{max}: Volume threshold for $T_{Volume,max}$ in packets or bytes
- $T_{notraffic}$: Each packet received during this period performs the gap calculation. Can be combined with $T_{Volume,max}$
- T_{Delay}: A time for which the expensive path in the scheduling process is not available, triggered by a gap size larger $T_{Volume,max} + T_{notraffic}$
- $T_{StartDelay}$: An initial one time delay after connection establishment for which the expensive path in the scheduling process is not available

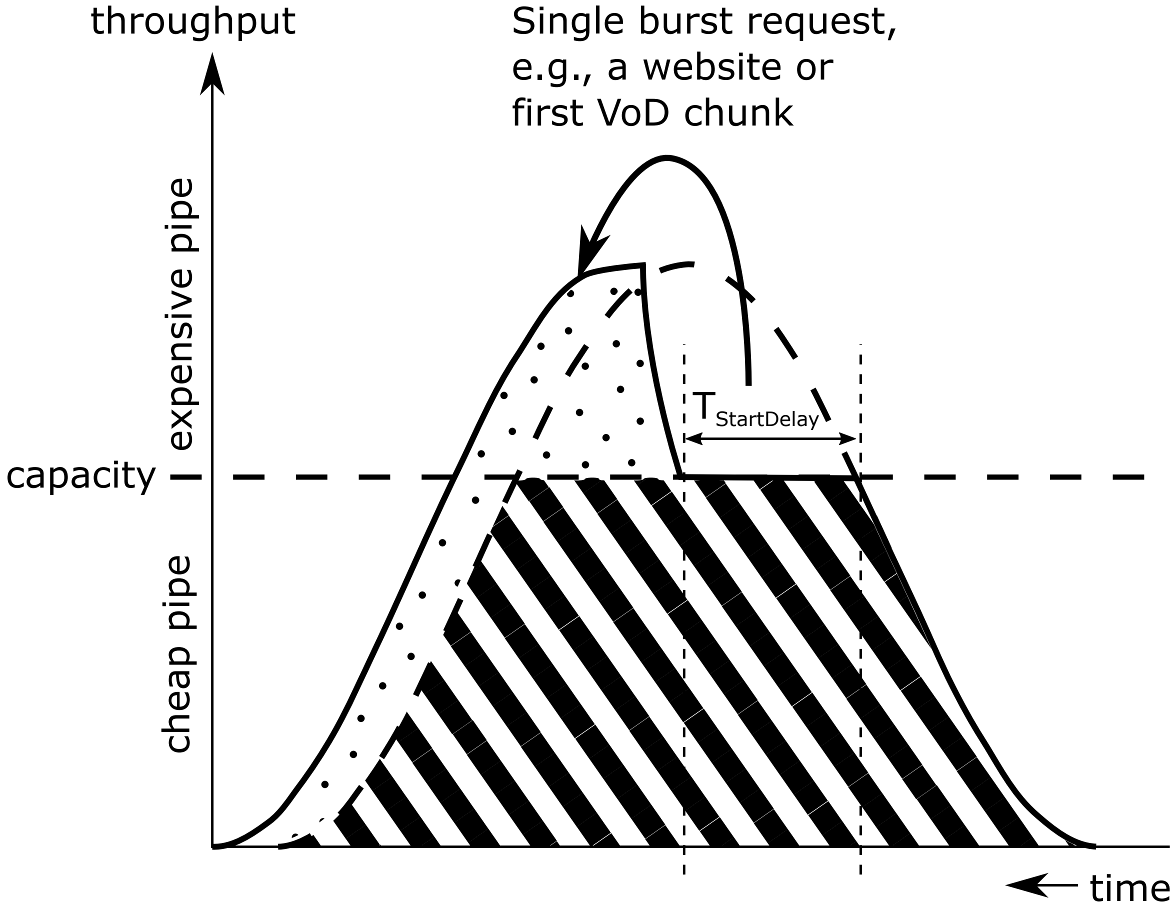


Fig. 3.13 Initial cost reduction after connection setup using $T_{StartDelay}$ without T_{GAP} measurement for single burst producing services or for the initial VoD chunk

Compared to the first idea of COM in subsection 3.5.3, the additional computational effort is limited to counting V_{max}. The determination of the gap size remains unchanged, since only the time between consecutive packets continues to be measured with the slightly modification that depending on $T_{Volume,max}$ a consecutive packet is accounted.

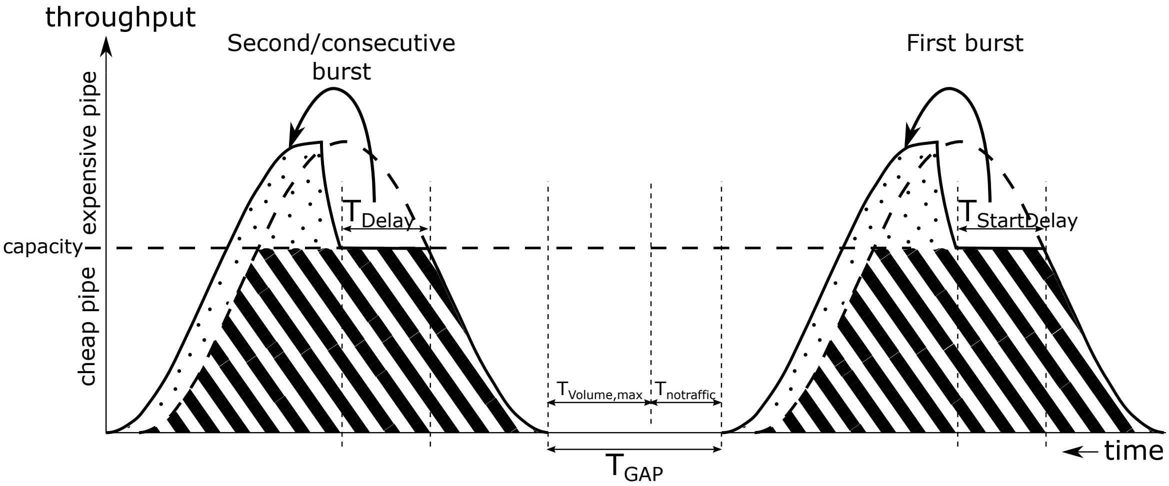


Fig. 3.14 COM- Practical idea to detect unsaturated link capacity based on the gap time T_{GAP} and T_{Delay} as measure to prevent spurious costly demand. $T_{notraffic}$, $T_{Volume,max}$ allow fine tuning of the gap detection, while $T_{StartDelay}$ optimizes the initial behaviour.

3.6 Solution summary and COM implementation

This section summarizes first the challenges *Cheapest-path-first (CPF)* scheduling faces when VoD traffic is present, second summarizes the development of countermeasures and third summarizes the description of the novel *COM* scheduler, which is able to alleviate the shortcomings of *CPF* scheduling. With these key essences of section 3.1 to section 3.5 the implementation of *COM* within this section and its relation to CPF is considered more tangible.

To understand better how a cost metric can be considered in a multipath system, some measurements were performed to assess the performance of the *CPF* scheduling logic in the presence of VoD traffic. For that, the MPTCP default scheduler decision logic[10] was modified to follow a simple linear traversal logic shown in Algorithm 1, which identifies the path p_i with the lowest cost C_i available – the non-exhausted send window – for the dispatch of a data segment. This implementation of a *CPF* scheduler also extends the Linux

[10]https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.95/net/mptcp/mptcp_sched.c#L195-L204

file system to store a cost indicator along with the network interfaces. From this, the *CPF* scheduler evaluates the cost of the paths provided by the MPTCP path manager along with the send window (*SWND*) information derived from the congestion control algorithm to verify the least cost path. The MPTCP open source prototype with its cross-version stable scheduler implementation is used in this work because of the protocol's maturity (as described in section 2.1) and practical relevance in the 5G ATSSS context (section 2.2. Because of MPTCP's strict in-order delivery inherited by TCP, focus can be put solely on exploration of scheduling without the impact of re-ordering, as long as buffer dimensioning follows [130] and it can be assumed that the application can sustain some latency which is typically the case if TCP is the selected transport protocol. In the same way, the *CPF* scheduler could be used for other congestion control based multipath protocols such as MP-DCCP and MP-QUIC. Even for the GRE-based Hybrid Access solution, *CPF* could be used if the remaining bandwidth of the fixed access path is monitored using the negotiated fixed access speed information instead of the *SWND* information.

Algorithm 1 (repeated)**:** CPF scheduler logic to return the least cost path for each dispatch of a data segment.

Initialization:
    $paths[] \leftarrow [[p_1, C_1], ..., [p_i, C_i]], \quad mincost \leftarrow max$

for $e \in paths$ **do**
    if $(e.C < mincost)$ AND $(SWND(e.p) > 0)$ **then**
        $bestpath \leftarrow e.p$
        $mincost \leftarrow e.C$

return $bestpath$

A measurement conducted using this implementation confirmed the suspicion expressed already in section 1.1 that bursty traffic produced by Internet dominating VoD services presents a major challenge for the *CPF* scheduling logic. In a setup similar to Hybrid Access, a MPTCP enabled home gateway and a MPTCP termination point, both running MPTCP Proxy, were connected using a commercial 6 Mbps DSL and a commercial cellular LTE connection. This setup is also the one used for demonstrating the purpose of this work and is further described in chapter 4. During the transmission of a 1080p VoD stream from an Internet VoD provider, 90% of the video was forwarded over LTE even if priority was on DSL, as shown in Figure 3.8. Compared to a single path transmission over DSL only, no benefit in terms of QoE was measurable as in both scenarios the video ran smoothly. Hence, the indication provided in Figure 1.2 is not misleading. This raises the question: **How can spurious demand be avoided if apparently no impact on the service delivery can be**

monitored? To counter the impression that this is a Hybrid Access phenomena, similar results were obtained when MPTCP and *CPF* scheduler was implemented in a smartphone.

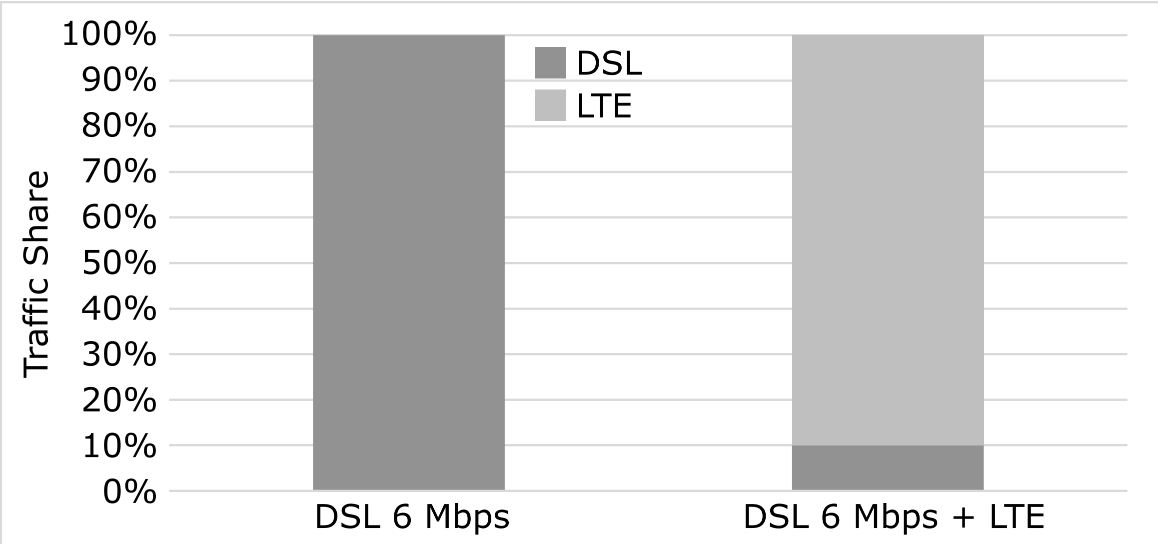


Fig. 3.8 (repeated) Traffic share for smooth HTTP VoD streaming with 1920x1080 H.264 over 10min comparing single path DSL and multipath DSL + LTE with MPTCP and *CPF* scheduler

These results demonstrate that a re-design of the scheduling logic is required to cope with the bursty nature of the VoD traffic. Ideally, the scheduler should avoid the usage of the high-cost path if the QoE for the user of a VoD application cannot be improved.

In order to define the design goals of the new algorithm, a look at the nature of traffic bursts in multipath scenarios should be taken, depicted in Figure 3.9. It can be noted that:

1. The overflowing part requires a costly transmission using the expensive path, and as such should be avoided.

2. There is a “valley” between the throughput bursts, leaving available capacity on the cheaper path unused.

Additionally, from the research in section 2.4 it is known that VoD within the scope of its receiver playout buffer capacity is delay tolerant.

With this in mind, design goals for the new algorithm can be defined as:

- the algorithm needs to be able to detect traffic bursts causing spurious demand in multi-connectivity scenarios

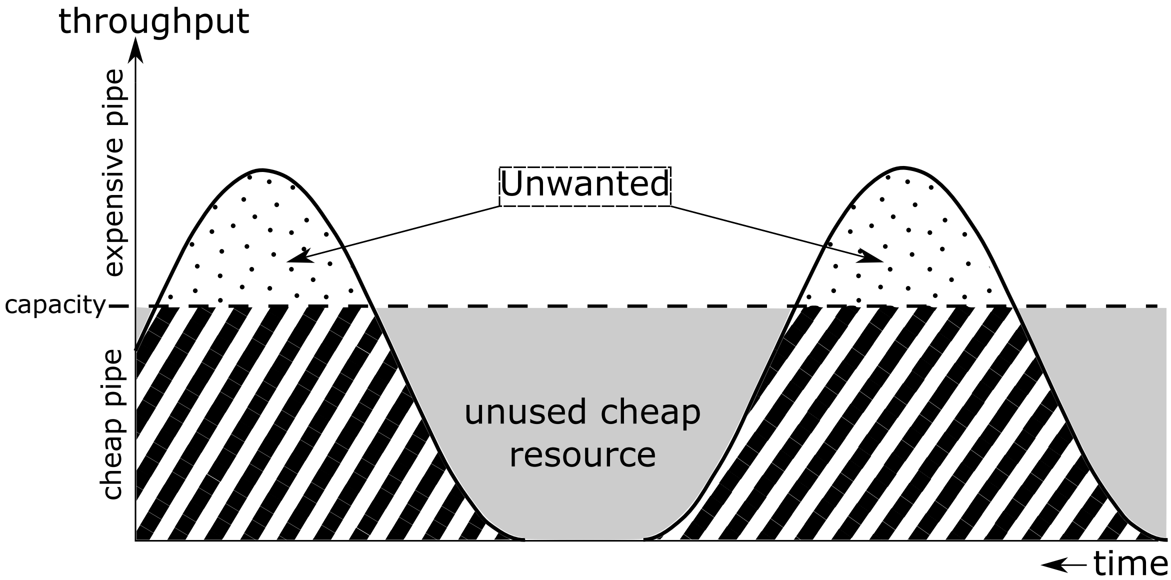


Fig. 3.9 (repeated) Unwanted multipath operation when traffic burst overflow into costly paths

- the traffic needs to be scheduled as much as possible on the cheaper resources, according to the real application and customer needs.
- the algorithm needs to be generic, simple, and service-agnostic, requiring no service-specific support
- the algorithm must not decrease the user QoE compared to single path transport but should result in a QoE similar to CPF.

Following Figure 3.9 and using B_{cheap} to denote the capacity of the cheaper resource and B_{demand} to denote the total bandwidth demand, it can be derived the following design principle for the new scheduling algorithm:

If Equation 3.7 is true, then prevent access to the expensive pipe.

$$\int_0^t B_{cheap} \geq \int_0^t B_{demand} \tag{3.7 revisited}$$

This new scheduler is called the *Cost Optimized Multi-path (COM)*. While Equation 3.7 provides many challenges to overcome, including dimensioning the time interval t, determining B_{cheap} and monitoring B_{demand}, an idea presented in Figure 3.14 can be used to design a working solution. Instead of monitoring capacities and demands, the T_{GAP} size can be used to identify the saturation information. To do this, the time gap T_{GAP} between consecutive packets can be measured and compared against a threshold value $T_{GAPthresh}$. If the condition given in Equation 3.8 is true, access to the expensive path is prevented for a time span of

T_{Delay}, with the multipath scheduler sending the packets to the cheaper path only. As a consequence of this, bursts will be stretched over time using a larger share of the cheaper pipe. A clear advantage is that COM does not require any further measurements besides T_{GAP}.

$$T_{GAP} \geq T_{GAPthresh} \tag{3.8 revisited}$$

In Algorithm 2 this principle is included and together with the *CPF* logic of Algorithm 1 forms the *COM* scheduler. Whenever the *CPF* scheduler foresees to send traffic over the low-cost path, the code from Algorithm 2 calculates T_{GAP} and blocks after verification of Equation 3.8 the expensive path or releases the path after the time T_{Delay}. During the phase where the expensive path is blocked, the *CPF* scheduler cannot select this path for dispatching data. This principle, with measuring the burst gap size, applies always to the path where traffic is currently scheduled to. In case a first low-cost path is saturated, *COM* will give access to a secondary mid-cost path. Providing a third high-cost path will be touched, if a saturation on the mid-cost path is measured, otherwise not. In a further logic not shown, the path blocking is also lifted when the low-cost path is no longer responsive due to a broken link. The variables that require preassigned values for the function of *COM* are listed in Table 3.1 and are initialised when the scheduler is engaged first in a TCP session. The initial values given in this table are indicative only and their determination is subject to chapter 5.

Algorithm 2: COM - Generic code logic for gap detection and overflow prevention into expensive path executed for each data to be sent on the low-cost path.

Initialization:
    $T_{now} \leftarrow now, \quad T_{GAP} \leftarrow T_{now} - T_{lastdata}$

if $T_{GAP} \geq T_{GAPthresh}$ **then**
    SET_BLOCK_FLAG(*expensive path*)
    $T_{block} \leftarrow T_{now}$
else if $T_{Delay} > (T_{now} - T_{block})$ **then**
    RELEASE_BLOCK_FLAG(*expensive path*)

$T_{lastdata} \leftarrow T_{now}$

As long as the achieved stretch of a burst does not let the VoD client's buffer run out of data, no additional access resources need to be used. The presence of T_{GAP} is an unmistakable sign that the buffer sufficiently re-fills, assuming that paths present in the multipath system have a Bandwidth Delay Product (BDP) including a safety margin for re-transmissions covered by the VoD client playout buffer. Typically, the BDP is not an issue in commercial networks, as services like Youtube or others smoothly run over DSL or LTE. Therefore *COM* is a self-regulated algorithm when it restricts access based on Equation 3.8. If an access

Variable	Initial value	Description
$T_{GAPthresh}$	e.g., 600 ms	Time threshold for detection of burstiness if time distance between consecutive packets - T_{GAP} - is greater or equal
T_{Delay}	e.g., 1000 ms	Time how long the epensive path is blocked in case of burstiness
$T_{lastdata}$	current time	Helper variable keeping the time of execution
T_{block}	0	Time of last block event

Table 3.1 Configurable parameters to be initialised at the start of COM

path provides a BDP that leads to the delivery of unusable – outdated – data, a mechanism is required to remove such paths from the multi-path scheduling. Since this is a general multipath problem that also affects CPF, it is implicitly taken into account when CPF and COM are later compared in the same access environments.

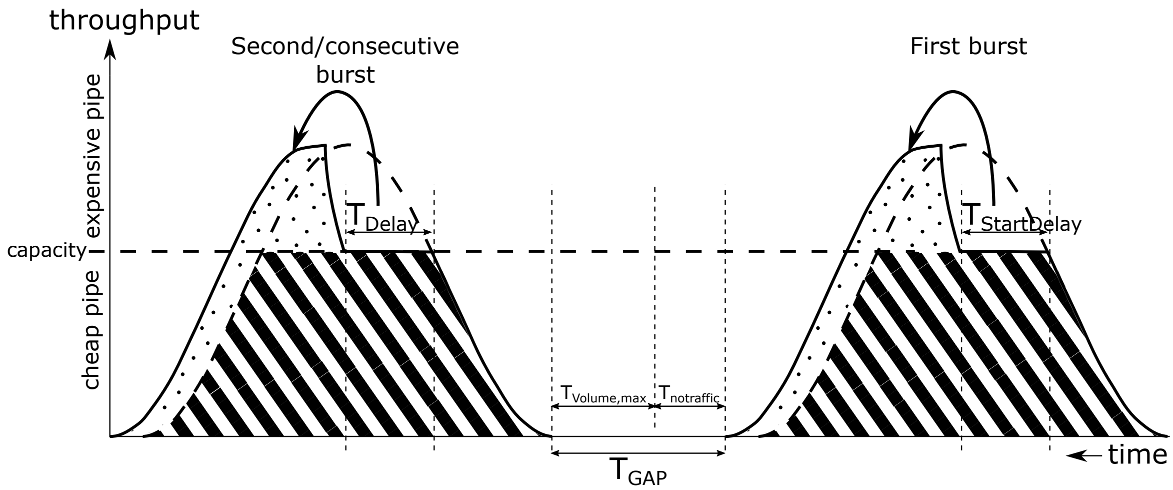


Fig. 3.14 (repeated) COM- Practical idea to detect unsaturated link capacity based on the gap time T_{GAP} and T_{Delay} as measure to prevent spurious costly demand. $T_{notraffic}$, $T_{Volume,max}$ allow fine tuning of the gap detection, while $T_{StartDelay}$ optimizes the initial behaviour.

The basic *COM* algorithm is as simple as maintaining three main variables namely the measured T_{GAP} and the configurable $T_{GAPthresh}$ and T_{Delay}. Figure 3.14, however, shows also the optional definition of $T_{notraffic}$, $T_{Volume,max}$ and $T_{StartDelay}$. With $T_{StartDelay}$ the first burst in a connection can be 'flattened' even if no T_{GAP} calculation could be carried out before. This might be useful to further optimize cost, but should be used carefully to prevent unwanted QoE degradation. According to the target traffic characteristic in Figure 3.9 the T_{GAP} calculation is in principle a good measure to detect spurious demand of VoD services, but any exchange of smaller amounts of data between bursts, e.g. control information or statistics between client and server, resets the calculation and makes *COM* assume high demand again. This type of exchange can be easily observed if, for example, a YouTube or Netflix video is paused during playback and a continuous exchange of data with URLs containing "stats" or "log" continues to take place. With the definition of $T_{notraffic}$ and $T_{Volume,max}$, T_{GAP} is split and becomes therefore more fine tuned. $T_{notraffic}$ is the time span in which any traffic will reset the gap calculation and corresponds in the absence of $T_{Volume,max}$ to the former T_{GAP}, otherwise T_{GAP} is represented by the sum of both and becomes less sensitive to smaller data exchange between bursts. This is achieved by allowing a certain amount of data within the time span of $T_{Volume,max}$ specified by the parameter V_{max}. Any data in the period of $T_{Volume,max}$ below the threshold of V_{max} does not lead to a reset of the gap calculation. For this, the algorithm remains able to distinguish between the VoD bursts without being confused by interfering data.

This enhanced logic of *COM* scheduler is shown in Algorithm 3 and aims to replace the code of Algorithm 2. In addition to the variables defined in Table 3.1, Table 3.2 lists the new variables for the finer granular gap detection. Similar to Table 3.1, the initial values given in this table are indicative only and their determination is subject to chapter 5. Also the Table 3.2 defines the $T_{StartDelay}$ from Figure 3.14, which executes `set_block_flag(expensive_path)` and `T_block = T_now` once after *COM* is initialized within a TCP session. For the presented code of the enhanced *COM*, V_{max} stands for the number of packets, but could also be used to define a volume if the calculation of V_{sum} takes into account the size of the data for scheduling.

Variable	Initial value	Description
$T_{notraffic}$	e.g., 50 ms	Time within $T_{GAPthresh}$ where transmitted traffic will reset the gap detection
V_{max}	e.g., 100 pkts	Max number of packets/volume during $T_{Volume,max} = T_{GAPthresh} - T_{notraffic}$ without resetting the gap detection
V_{sum}	0	Volume counter during $T_{Volume,max}$
$T_{StartDelay}$	e.g., T_{Delay}	Block time of the expensive path after COM initialization

Table 3.2 Configurable parameters to be initialised at the start of enhanced COM in addition to Table 3.1

To get a complete picture, the flowchart of the COM dispatch logic is shown in Figure 3.15. Just like the *CPF* flowchart in Figure 3.2, the scheduler loops over the available MPTCP subflows to determine the `best` path. The use of `min_prio` and `min_srtt` is the same and is therefore omitted. Also, the short form `prio` and `srtt` is the same as the long form chosen in Figure 3.2 with `prio(subflow)` and `srtt(subflow)`. Instead of one socket pointer `sk` in Figure 3.2 which holds and returns the “best” path available which has the lowest cost (lowest prioritization value), the *COM* scheduler defines two socket pointers

Algorithm 3: Enhanced COM - Generic code logic for tolerant gap detection and overflow prevention into expensive path executed for each data to be sent on the low-cost path.

Initialization:
    $T_{now} \leftarrow now, \quad T_{GAP} \leftarrow T_{now} - T_{lastdata}$

if $T_{Delay} > (T_{now} - T_{block})$ **then**
    RELEASE_BLOCK_FLAG(*expensive path*)

if $T_{GAP} \geq T_{notraffic}$ AND $T_{GAP} < T_{GAPthresh}$ AND $V_{sum} < V_{max}$ **then**
    $V_{sum} \leftarrow V_{sum} + 1$
    goto *end*

if $T_{GAP} \geq T_{GAPthresh}$ **then**
    SET_BLOCK_FLAG(*expensive path*)
    $T_{block} \leftarrow T_{now}$

$V_{sum} \leftarrow 0$
$T_{lastdata} \leftarrow T_{now}$

end:

`best` (used like `sk` for *CPF*) and `best-blocked`.

As long as no path is marked as `blocked`, which is controlled by the *COM* gap detection algorithm in Algorithm 3, *CPF* and *COM* will deliver the same result – `return best`. However, if an expensive path is `blocked` because the gap detection algorithm recognized on the cheaper path a $T_{GAP} > T_{GAPthreshold}$, the expensive path is under normal circumstances not available for scheduling. In this case, when the high cost path is blocked, no data can be scheduled if the low cost path is congested – `return NULL`. `return best` then makes the lowest cost path available again only when the low cost path becomes available again or the blocking flag of the expensive path is released. So far, this simply follows the *CPF* design with the small difference that an expensive path might be removed from the scheduling process and is not available to transmit data if the low cost path is not available.

A special case is treated by the `any_path` variable which avoids a transmission stall if all non-blocked paths are broken, e.g. due to connection loss. In this case `return best-blocked` ensures transmission continuation and provides the uncongested path with the lowest cost out of the list of paths with the `block` flag set. To enable this function, the `subflow available` check in Figure 3.2 is separated into its two components of checking for broken path and detecting congestion ($SWND == 0$?). This facilitates a more granu-

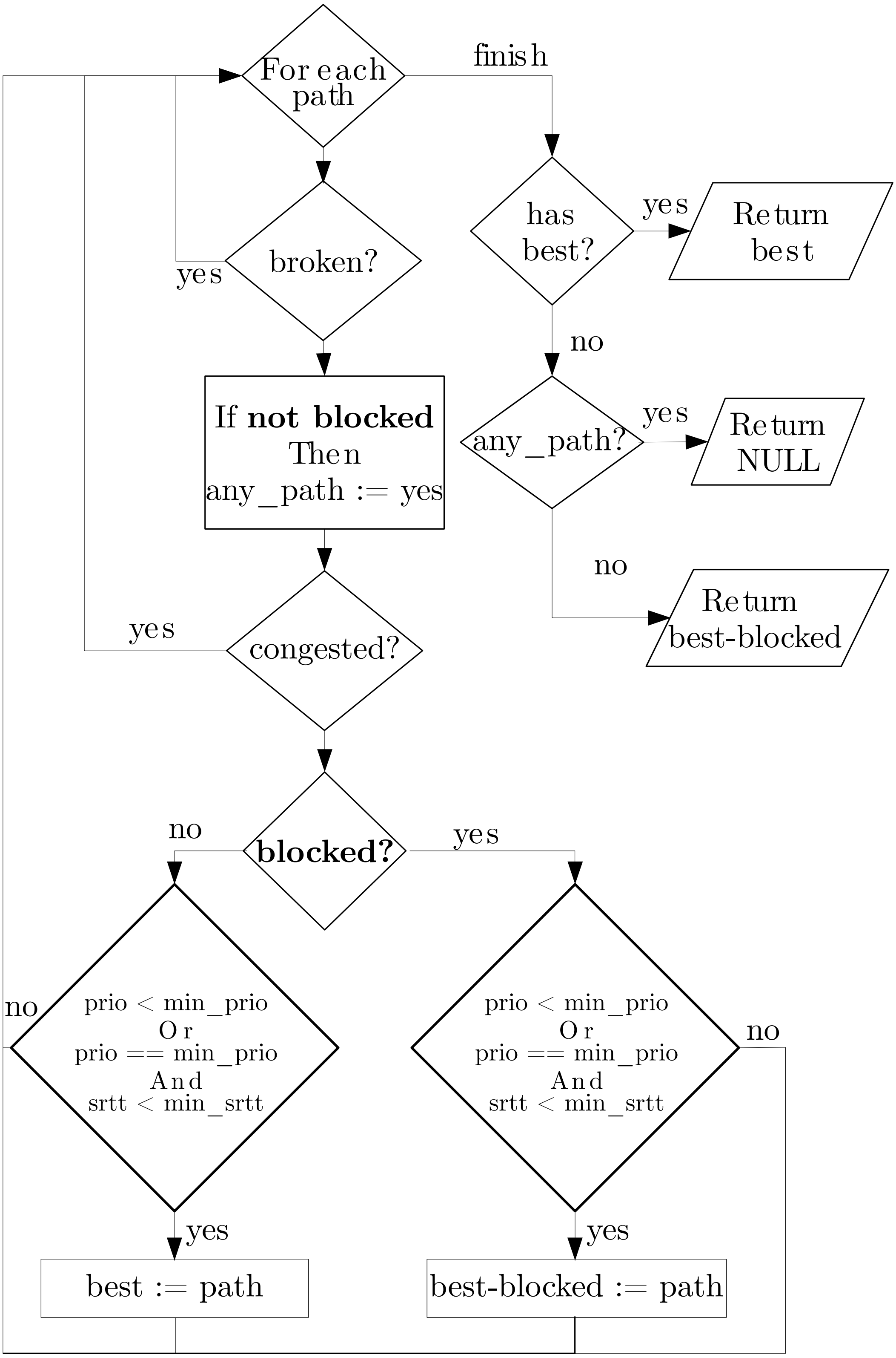


Fig. 3.15 Flow diagram of the MPTCP COM scheduler dispatch logic

lar processing of a path unavailability and helps to distinguish between a `broken` and a `congested` path. This finally allows to identify if at least one `non-blocked` path is available to continue the default *COM* scheduling procedure or if `blocked` path needs to be used instead to avoid transmission stall.

Similar to the description of the *CPF* parametrization through the Linux `sysfs` in subsection 3.2.3, *COM* can be controlled. For this, the new files according to the parameters of `mptcp_t_gapthresh`, `mptcp_t_delay`, `mptcp_t_startdelay`, `mptcp_t_notraffic` and `mptcp_t_v_max` are added to the `/sys/class/net/*` tree and will be configured as defined in Table 3.3. A possible usage of the required parameters for *COM* is shown in Listing 3.4 including the required basic *CPF* parametrization already defined in Listing 3.2. The parametrization design follows the idea to configure all relevant *COM* parameters on the expensive access interface, although `mptcp_t_gapthresh`, `mptcp_t_notraffic` and `mptcp_t_v_max` are monitored on the MPTCP subflow which refers to the next lower priority access.

Sysfs variable	Unit	Description
`mptcp_t_gapthresh`	ms	Overall time threshold for the detection of burstiness covering `mptcp_t_notraffic + mptcp_t_v_max`
`mptcp_t_delay`	ms	Block time of the expensive path after burstiness detection
`mptcp_t_startdelay`	ms	Block time of the expensive path after COM initialization
`mptcp_t_notraffic`	ms	Time within $T_{GAPthresh}$ where transmitted traffic will reset the gap detection
`mptcp_t_v_max`	no. packets	Allowed packets in the time `mptcp_t_gapthresh - mptcp_t_notraffic` without

Table 3.3 Explanation of Linux sysfs COM variables

```
# Basic CPF settings
echo 0 > /sys/class/net/wlan0/mptcp_prio
echo 1 > /sys/class/net/wwan0/mptcp_prio

# COM settings expensive path
echo 400  > /sys/class/net/wwan0/mptcp_t_gapthresh
echo 4000 > /sys/class/net/wwan0/mptcp_t_delay
echo 4000 > /sys/class/net/wwan0/mptcp_t_startdelay
echo 50   > /sys/class/net/wwan0/mptcp_t_notraffic
echo 100  > /sys/class/net/wwan0/mptcp_v_max
```

Listing 3.4 Exemplary COM parameterization through Linux `/sys/class/net/` tree

As shown in subsection 3.2.4 *CPF* also handle more than two paths, which is only limited by practical considerations like available accesses, system resources or expected performance gain. *COM* as an enhancement of *CPF* does not change this principle. The obvious parametrization design principle discussed above, also allows to configure paths beyond the two paths. Listing 3.5 shows this for a low cost Wi-Fi, an expensive cellular access and a most expensive satellite link. While the *COM* settings defined for the expensive cellular path monitor the gap development on the Wi-Fi path, the setting for the satellite link monitor this evolution on the cellular path. The resulting path selection of the COM scheduler depending on the possible parameters cost configuration, path state and COM determined temporary blocking of the path can be found in tables in the Appendix A for a two and three path system.

Even though it was verified during the implementation that COM handles more than two paths, the evaluation of this is out of the scope of this work due to lack of relevance. Both the HA and the 3GPP ATSSS Rel. 16 are not known to use more paths in practice or even to limit themselves to this number in the specification.

In contrast, the impact of the measured T_{GAP} variable along with the configurable optimization parameters $T_{GAPthresh}$ and T_{Delay} defined in Table 3.1 and their companions for more fine granular optimization control $T_{notraffic}$, V_{max} and $T_{StartDelay}$ defined in Table 3.2 are discussed as part of the evaluation in chapter 5.

```
# Basic CPF settings
echo 0 > /sys/class/net/wlan0/mptcp_prio
echo 1 > /sys/class/net/wwan0/mptcp_prio
echo 2 > /sys/class/net/sat0/mptcp_prio

# COM settings expensive path
echo 400  > /sys/class/net/wwan0/mptcp_t_gapthresh
echo 4000 > /sys/class/net/wwan0/mptcp_t_delay
echo 4000 > /sys/class/net/wwan0/mptcp_t_startdelay
echo 50   > /sys/class/net/wwan0/mptcp_t_notraffic
echo 100  > /sys/class/net/wwan0/mptcp_v_max

# COM settings most expensive path
echo 400  > /sys/class/net/sat0/mptcp_t_gapthresh
echo 4000 > /sys/class/net/sat0/mptcp_t_delay
echo 4000 > /sys/class/net/sat0/mptcp_t_startdelay
echo 50   > /sys/class/net/sat0/mptcp_t_notraffic
echo 100  > /sys/class/net/sat0/mptcp_v_max
```

Listing 3.5 Exemplary COM parameterization through Linux `/sys/class/net/` tree using three access paths

3.7 Impact of COM on VoD and other traffic

The idea of *COM* is to be applied as a permanent replacement of the *CPF* without the requirement of service or traffic classification. The simplicity of the *COM* algorithm makes it possible to make some assumptions on how it behaves under certain traffic scenarios, especially if there is an immediate and comprehensive need for the full aggregated throughput (e.g. file download). In this context, one can observe the following use cases:

- **VoD with** $B_{demand} < B_{cheap}$
  Due to the original bursty nature of the video data, a time gap should be visible between the traffic bursts end-to-end, as long as no intermediate bottleneck disrupts this. If *COM* can monitor this, it can schedule data to the cheaper path and access to the expensive path is not required.

- **VoD with** $B_{demand} > B_{cheap}$
  Different to the use case above, demanding a higher throughput than the cheaper

pipe can provide will constantly fill this path and no gap will be detected. Access is therefore given to the expensive path which is now responsible to drain the overflowing traffic.

- **VoD with adjustable resolution**
  It is expected that this will also match the case when a VoD service dynamically adjusts the video resolution according to the available throughput. Such a situation will lead to either the first or to the second use case. At least an upgrade to a higher resolution should not be blocked, since the gap will become shorter or even vanish.

- **File download**
  A constant file download should not be affected at all. This kind of traffic is out of scope of this work since it already works with the *CPF* scheduler, as demonstrated in [21]. In the case the scheduler verifies a constant demand on the cheaper path, justified by the nature of a file download without gaps, access is provided to the expensive path. It also does not matter if the file download demand is below or above the capacity of the cheaper pipe, as the basic *CPF* principle kicks in.

- **Bottleneck before the scheduler**
  In the case the bottleneck is not the cheaper path and the bottleneck appears before the traffic reaches the multipath scheduler, the **file download** use case is applied.

As specified in section 3.5, the File Download case will apply whenever $\frac{Packetsize}{Bitrate} < T_{GAPthresh}$. In a scenario with a packet size of $\sim 1500\,\text{B}$ corresponding to a typical specified Maximum Transmission Unit (MTU) of a network link, a new packet is scheduled every 120 ms at a bit rate of 100 kbps and 12 ms at a bit rate of 1 Mbps without taking jitter into account. If a $T_{GAPthresh}$ is above these values, a file transfer is reliably recognised and if not, the transfer rate is so low that any DSL connection can cope with the rate itself without an additional path.

These assumptions are strengthened by the key result of a field trial with 30 customers using *COM* and *CPF* subsequently for any traffic exchange with the Internet on multipath enabled smartphones. The accompanying customer surveys did not reveal any degradation of the perceived QoE if *COM* was applied, while transmission cost clearly went down compared to *CPF*. More details on this can be found in the relevant section 5.6.

3.8 COM scheduler within 5G ATSSS

A main implementation scenario considered for *COM* is its usage in 3GPP ATSSS for access aggregation. In section 2.2 the three specified scheduling mechanisms in ATSSS are analyzed and the identified steering mode relevant for this work is the *priority based* one which allows the prioritization according to a cost metric. Its definition in [11] says:

Priority-based: It is used to steer all the traffic of an SDF to the high priority access, until this access is determined to be ***congested****. In this case, the traffic of the SDF is sent also to the low priority access, i.e. the SDF traffic is split over the two accesses. In addition, when the high priority access becomes unavailable, all SDF traffic is switched to the low priority access. How UE and UPF determine when a congestion occurs on an access is implementation dependent. It can only be used for the non-GBR SDF.*

This is equivalent to the *CPF* principle. Similarly, the *COM* meets the requirements of the *priority based* steering function. Compared to the *CPF*, however, the bold marked term "congested" is interpreted differently. While in *CPF* the TCP send window is solely used to identify congestion, *COM* uses in addition the size of the "valley" between bursts.

This leads in the next step to the consideration of the steering functions which needs to support access aggregation and prioritization. In Figure 3.16 both steering functions are depicted with MPTCP highlighted in red as the only one able to fulfill the requirement. The other steering function, ATSSS-LL, is not designed for this case because there is no means to detect an exhausted prioritized path to switch traffic to a non-prioritized path.

Following these findings, it can be noted, that both, the *CPF* MPTCP implementation developed in subsection 3.2.3 and the *COM* implementation described in subsection 3.5.4 can be used without modification for the purpose of *priority-based* steering in ATSSS. This statement becomes clearer if the 3GPP architecture is harmonized with the multipath system model (Figure 3.1) developed in this work.

For a better understanding, the model of Figure 3.1, which served as the basis for the development of *CPF* and *COM*, is extended by the terminology and entities of the 3GPP architecture in Figure 3.16. Essentially, it is the naming of the access paths 3GPP and non-3GPP and the use of MPTCP as the responsible multipath network protocol that results in a practical specification. Compared to Figure 3.1, the naming of the path can be simply translated into a first and a second path, while MPTCP provides the typical components

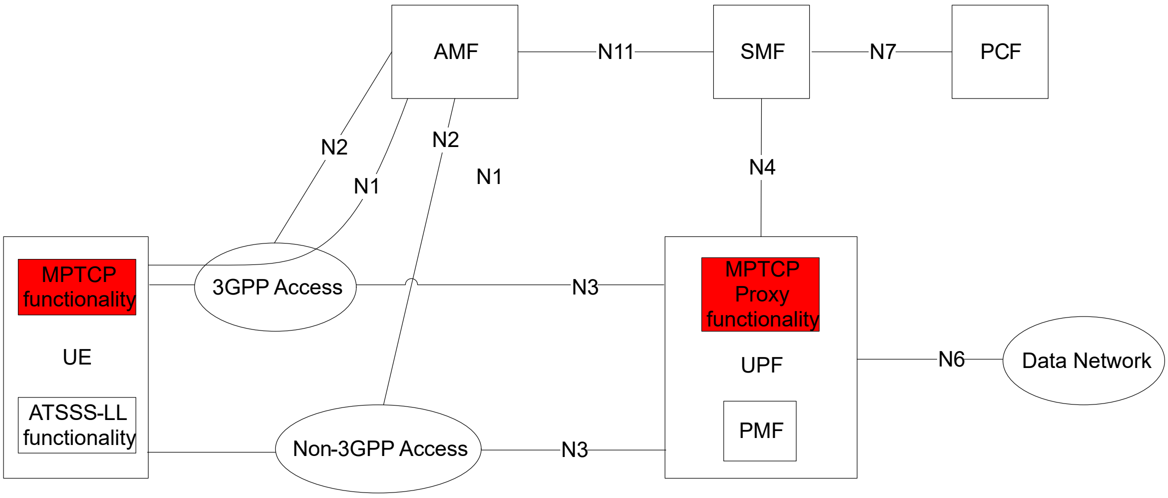


Fig. 3.16 5G system with ATSSS architecture in the non-roaming scenario and untrusted non-3GPP access [11]

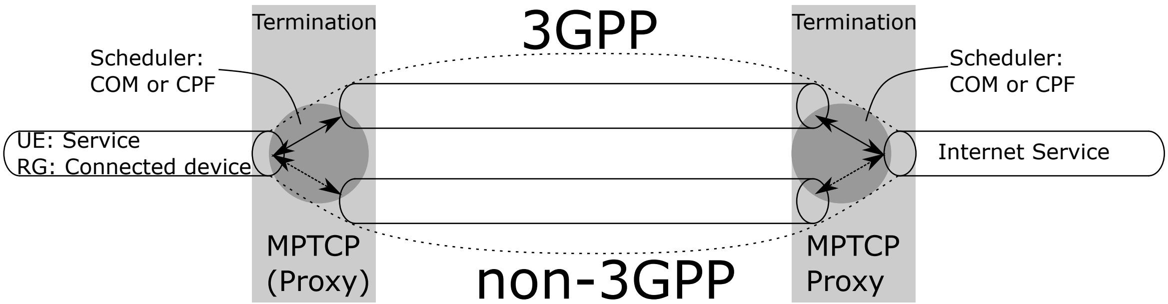


Fig. 3.17 Replication of the used multipath system model into the ATSSS architecture

required for multipath transport as described in Figure 1.1, including the multipath scheduler.

Depending on the scenario HA/WWC (Figure 2.2) or ATSSS (Figure 2.1) the MPTCP is proxied in both termination points or only on one side. The proxying mechanism, which is just a mean to translate between MPTCP and TCP traffic to achieve end-to-end transparency, is further discussed in section 4.1 to develop appropriate test environments. However, the central point is the multipath transport between the both MPTCP entities which corresponds to the underlying system model of this work. An evaluation of this setup with the *CPF* or *COM* scheduler, both of which match the *priority-based* steering mode of ATSSS, will provide results that can also be expected from a commercial ATSSS or HA deployment.

Chapter 4

Methodology and testbed

With the development of the algorithm and the analysis of use cases, COM looks quite promising on paper. The attractive idea of using the time gap between packets to assess demand requires only a small change to the CPF scheduler. Now the next step is to take practical measurements to prove the cost effectiveness compared to CPF without compromising the customer experience.

In order to evaluate the performance of the COM scheduler, firstly a methodology needs to be developed to explore COM's impact and secondly, a suitable testing environment is required.

There is a clear expectation associated with the use of multipath transport that it will compensate for the weaknesses of singlepath transport and thus deliver better, or at least not worse, QoE. However, the preliminary results presented in Figure 3.8 lead to the assumption that CPF has under certain scenarios no benefits for the service and just produces spurious cost when the high cost path is used. A solution space for COM develops thereof in the space presented in Figure 4.1, which shows the area of tension in a multipath system between the additional cost required and the achievable QoE gain. The dimension of the system is clearly defined by the *Non-Aggregation* (NA) operation point in the origin $O(0|0)$ and the *Unlimited-Aggregation* (UA) operation point $P(max|max)$. By this definition for a given transmission scenario, *NA* represents the single low-cost path performance, while *UA* represents the multipath system where any path can be used as desired as for example CPF scheduler implements in the Hybrid Access scenario used to initially demonstrate the cost contradiction in section 1.1. If one take the experiment in Figure 3.8 as an example, this results in a solution space between $O(0|0)$ – DSL only test – and $P(\rightarrow max|0)$ – multipath with CPF test – since the QoE has not changed, but the cost.

Therefore, it follows that the performance of COM must prove itself against the Hybrid Access (HA) scenario with CPF (*UA*) and the corresponding single low-cost path transport

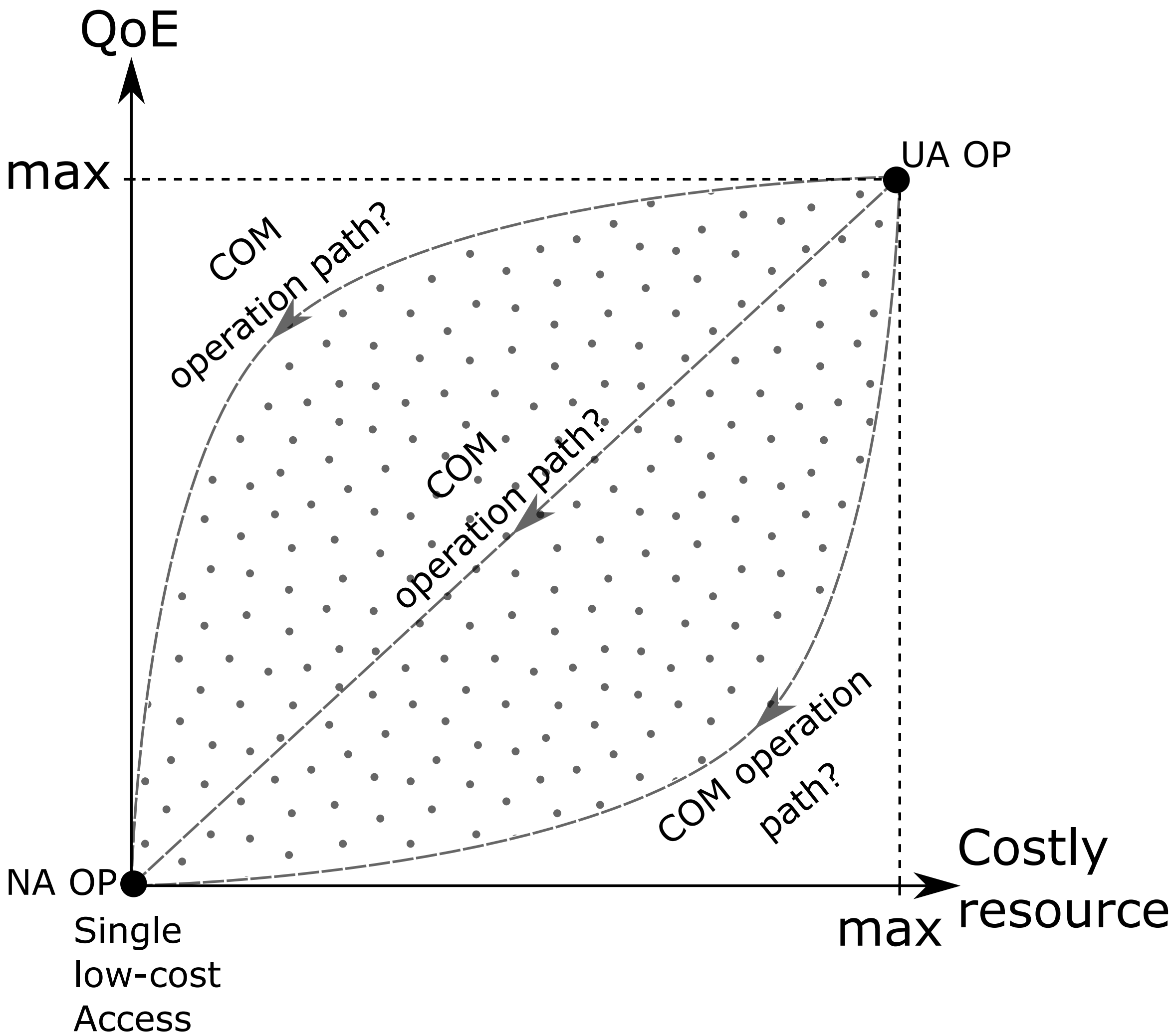


Fig. 4.1 COM solution space considering QoE and costly resource consumption

scenario (*NA*), e.g. DSL only, which seems in some cases to already provide the best QoE as shown in Figure 3.8. Furthermore, if COM proves to work in the HA scenario using a RG, it is natural to also investigate the ATSSS UE case.

In the light of Figure 4.1 the verification of COM's operation path is the main testing objective with the following general principle: *Moving the today's Hybrid Access UA operation point within the solution space and minimizing the access to the costly resource while keeping the QoE at a sufficient level.*

Traversing the Figure 4.1 solution space requires a testbed equipped with the MPTCP scheduler implementation of COM in combination with the use of an VoD service as the main optimisation goal of this work, but also suitable for testing non-VoD traffic to ensure safe interaction with this traffic. The different characteristic of testbeds developed in the course of this work and their purpose are discussed in section 4.1.

In section 4.2 a comparative approach is developed that provides a QoE gain and a cost factor based on the calculation of different parameters of QoE and cost to analyse the performance of COM over singlepath and CPF.

The measurements that need to be collected to finally evaluate the performance of COM and how to obtain the parameters in the systems equipped with the MPTCP scheduler implementations of this work is the purpose of section 4.3.

4.1 Testbed

As a consequence of the solution space to be explored in Figure 4.1, the high-level testbed architecture in Figure 4.2 results. It shows a typical Hybrid Access (HA) scenario that corresponds to the architecture in which the inefficiency of CPF scheduling was first detected (Figure 1.2). This is the baseline scenario in which COM has to prove its performance, while more specific details and other scenarios used in this thesis are elaborated in subsection 4.1.1, subsection 4.1.2 and subsection 4.1.3. Based on the ATSSS and WWC architecture from Figure 2.2 it maps the multipath scheduling relevant entities 5G-RG and UPF and MPTCP as the relevant network protocol. Here, a Hybrid Access Router (5G-RG) provides residential connectivity and communicates over DSL and LTE – both permitted access types in WWC, although the entity is called 5G-RG – with a Proxy (UPF) entity to terminate the multipath transport and provides the gateway for Internet access. The decision to use LTE instead of 5G is due to the lack of 5G connectivity at the test site. However, this has no impact on the evaluation, as specification of LTE and 5G characteristics for Mobile Broadband (MBB) connectivity only show differences in the peak data rate. Furthermore, the focus of COM is more on the traffic pattern and gap detection over the cheaper path, making a possible small performance difference between LTE and 5G – both expensive paths – insignificant for validation.

With the implementation of MPTCP and a TCP-Proxy on Router and Proxy entity, as required by the ATSSS specification, TCP communication between services originating in the Internet and clients behind the router, are transparently enabled for multipath transport. Both, Router and Proxy needs to implement for the purpose of this thesis the COM and CPF scheduler as per Algorithm 3/Figure 3.15 and Algorithm 1 and must be be configurable to apply one of both or to apply transport over the single cheap path. It must be noted, that the scheduler located in the Proxy takes care of the downlink traffic from Internet to client and the multipath schedulerimplemented in the Router for the uplink traffic from client to Internet.

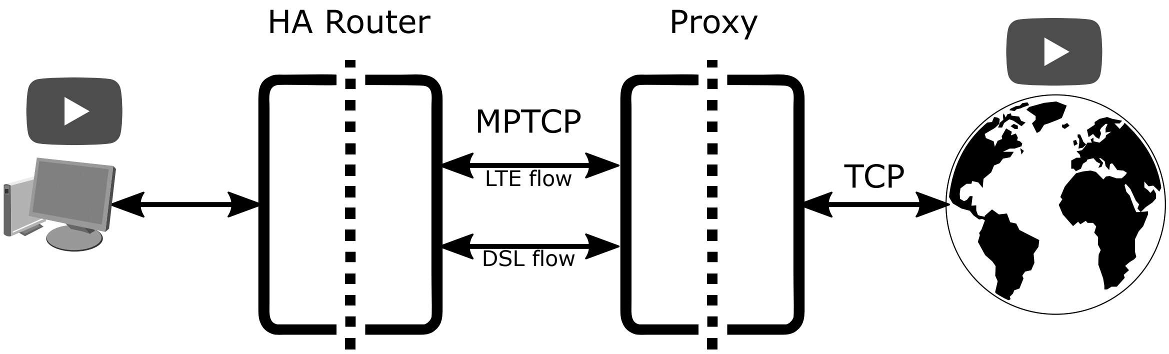


Fig. 4.2 MPTCP testbed for verification of COM impact over online services

As part of this work, various test environments were developed and used to study the impact of COM during the various steps from initial design to a multi-month trial with customers of an Mobile Network Operator (MNO).

At the beginning of the implementation phase of CPF (subsection 3.2.3) and COM (subsection 3.5.4) in MPTCP, a local testbed, which is further described in subsection 4.1.1 has provided valuable services with maximum control over all features and behaviors

As COM, defined by the objectives of this work, can only be considered useful if it works in a transparent multipath scenario and can cope with traffic generated by non-controlled OTT VoD service provider, the Internet connected testbed detailed in subsection 4.1.2 introduces a multipath proxy functionality. While this is still an intermediary step towards the all-encompassing ATSSS/HA architecture this testbed created the confidence that the proxy does not have a noticeable impact on the multipath scheduler performance. For example, the test results collected in this test environment led to the development of the more robust gap detection algorithm explained in subsection 3.5.4.

For the purpose to evaluate COM in such a way that it allows conclusions to be drawn about its use in ATSSS and HA, the testbed in subsection 4.1.3 was ultimately designed. Its implementation resembles the key characteristics of the ATSSS architecture for UE multipath and the ATSSS based HA architecture for residential multipath analyzed in section 3.8. This testbed is able to deal with OTT VoD transmission but is also able to carry any other traffic. In addition, it enables the collection of all data required for the assessment and thus forms the basis for all assessment results presented in chapter 5.

Before the different testbeds are further clarified, the focus is first on the commonalities. All testbeds employ the MPTCP Linux reference implementation provided at https://multipath-tcp.org. Due to the different testbeds and operating times, different versions between 0.90 and 0.95 were used. The reason for this was mainly because of the different Linux versions supported by them: 3.18, 4.1, 4.4, 4.9, 4.14, 4.19. Especially for the implementation in an Android smartphone, it simplifies the integration of MPTCP if the Linux Kernel of the Android device and that of the MPTCP Linux reference implementation, overlap or are close to each other. The selection of the versions[1] between 0.90 and 0.95 did not require changes to the MPTCP scheduler design developed in this work. The MPTCP internal interfaces and the behavior behave cross-version stable. For any testbed COM and CPF was merged into the MPTCP Linux reference code and therefore built into the Kernel. The principle setup with two links follows the recommended routing configuration[2] with the default gateway pointing to the access interface related to the cheap path and merely adapted to the IP address configuration of the respective scenario. Also the activation of MPTCP follows the guidelines of the maintainers provided [3] and allows therefore the simple configuration and re-configuration of the setup for the testing with CPF, COM and Single Cheap Path (SCP). The detailed configuration of CPF for the cost metric and that for the parameters of COM can be found in subsection 3.2.3 and 3.6 and are inspired by the handling of the configuration interface of the MPTCP reference implementation.

With such a prepared testbed the deactivation of MPTCP with the Linux `sysctl` command:

```
sysctl -w net.mptcp.enabled = 0
```

causes the usage of SCP. Contrary, the activation of MPTCP with

```
sysctl -w net.mptcp.enabled = 1
```

enables the hooking into any new established TCP connection using the service transparent handshaking and subsequent TCP subflow procedure described in [10]. By default the *minRTT* scheduler is in use if not specified otherwise. With

[1]https://github.com/multipath-tcp/mptcp/tags
[2]https://multipath-tcp.org/pmwiki.php/Users/ConfigureRouting
[3]https://multipath-tcp.org/pmwiki.php/Users/ConfigureMPTCP

```
sysctl -w net.mptcp.mptcp_scheduler = cpf
```

or

```
sysctl -w net.mptcp.mptcp_scheduler = com
```

CPF or *COM* can be configured which then uses the information stored along with the access interface (CPF: Listing 3.2, COM: Listing 3.4) for the parametrisation of the scheduler. It shoud be noted whenever COM is used for measurement, $T_{StartDelay}$ is set to the value of T_{Delay}. This helps to reduce the number of test cases and is rational since a well working identified T_{Delay} can be used to investigate its impact on the QoE related to the connection establishment phase.

This enables comfortable use of the testbeds, as it allows changes to be made on the fly without time-consuming reboots or the like.

In a similar way the congestion control is configured using the Linux built-in CCs.

```
sysctl -w net.ipv4.tcp_congestion_control = bbr
```

or

```
sysctl -w net.ipv4.tcp_congestion_control = cubic
```

are used to verify the co-existence of multipath scheduler with the most common CCs used for VoD transmission, Cubic and BBR. Both will help to assess if the evaluation across CCs persists as discussed in section 2.3 and obsoletes tests with other CCs. Another benefit of selecting these two CCs are the verification of the interworking with the traditional ACK-clocked mechanism used by Cubic – Windows/Linux/Mac default CC – and the BDP optimization of BBR – used mainly in Alphabet universe (e.g. Youtube). Both CCs cover the great majority of VoD applications today.

Another important functionality is the MPTCP path manager which is responsible to trigger the subflow creation. Without a path-manager defined, no subsequent subflow creation occurs which leads MPTCP ad absurdum. The most common used path-manager is the `fullmesh` which creates a full-mesh of subflows across the available subflows and results in a $E \times F$ mesh with E client addresses and F server addresses. These addresses are learned when a new MPTCP connection is initiated using `MP_CAPABLE`, a subsequent flow is established using `MP_JOIN` or `MP_ADDADDR` and `MP_REMOVEADDR` announce new or outdated addresses. In case both entities, client and server, are equipped with two interfaces, two subflows are created by the `fullmesh` path manager per interface. This might be desired effect in an end-to-end multipath communication, but for the purpose of investigating CPF and COM for ATSSS and HA this is not seen as beneficial. Also it might raise side effects like inter-path competition by congestion control. To avoid such effects the **full-mesh** is the selected path-manager for the testbeds, but firewall rules are used to restrict the number of subflows per network interface to one.

The testbeds enable first and foremost tests with both static and adaptive resolution videos, but are not restricted to this scenario and also offer the possibility to test any type of traffic or service. A primary focus is on the verification with Youtube as the leading VoD service. Pre-tests with other VoD applications from *Vimeo* or *hls.js* confirmed that those services all rely on the same transmission technologies DASH or HLS using the transmission principle outlined in section 2.4. Their interaction with COM can be therefore considered to be covered when tests with Youtube are conducted.

Due to the focus on ATSSS and HA the number of paths for the evaluation is set to two. This is an artificial limitation not conditioned by MPTCP, CPF/COM or the testbed design. However, it is in line with the objective of this work and avoids the generation of unnecessary results.

The network links used in the testbeds always use a cost logic of a cheap link and a costly link. In testbeds which employ a cellular link this link always corresponds to the costly path. For the exploration of scheduling algorithm the performance, as outlined above, is compared between singlepath transmission and multipath transmission using CPF and finally COM. With the default gateway settings described above, it is ensured, that in case of single path transmission the cheap path is used (SCP). Links with different throughput on the cheap path make it possible to vary the characteristic of the cheap path, which is used by the schedulers CPF and COM as the main input but interpreted differently to estimate when it is necessary to use the expensive path for data transmission.

For interpretable measurement results certain things are always ensured to avoid arbitrary side-effects:

1. With the hardware selected for the devices under control

    - MPTCP client/server

- RG, UE or MPTCP Proxy
- Service consuming device

the resource like CPU, were dimensioned to not create a resource bottleneck which interferes with the results. This was verified with the devices in Table 4.1 in their various functions under the maximum throughput conditions applicable to the test scenario.

2. TCP buffer settings were selected according to [130] covering the expected sum bandwidth and latencies. Unless specified otherwise, `tcp_rmem=tcp_wmem=4096 1048576 10048576` defines in all testbeds a maximum TCP send and receiver buffer of 10MB on the MPTCP termination points.

3. The respective MPTCP traffic scheduler and configuration under test is always deployed on each side of the MPTCP termination points.

4. Any traffic on UDP port 443 is blocked:
   During the measurement phases of this work, QUIC established as alternative to TCP and became the preferred transport protocol for Youtube in Android environments. The almost always reliable method to suppress any QUIC and trigger a fallback to TCP at this time was to block the most common port to be used for QUIC. The measurements showed a high success rate of this method and a return to TCP for 99% of the traffic, which was important for the significance of the field test carried out in section 5.6..

Device	HW details	Usage
PC Engines APU2	AMD GX-412TC, 4 Core @ 1 GHz, 4GB DDR3-1333 ECC, 4x1Gbps Ethernet	RG, Proxy
KaToM-5006LP	Intel Core i7-6700 Socket 1151, 3.4GHz, 4core, DDR4, 8 GB,SODIMM, 2133 MHz, 6x1Gbps Ethernet	Consumer Device, MPTCP Client, RG, Proxy
Dell PowerEdge R640 DX152 (Datacenter Central Germany)	2x Intel® Xeon® Silver 4114 (10 Core @ 2.20 GHz, Skylake), 64GB DDR4 ECC, 10Gbps Uplink	Proxy

Google Pixel 2	Qualcomm Snapdragon 835, 4 GB LPDDR4X RAM, Qualcomm Snapdragon X16 LTE, Wi-Fi 5 (a/b/g/n/ac) 2.4 + 5.0 GHz	UE
Google Pixel 4	Qualcomm Snapdragon 855, 6 GB LPDDR4X RAM, Qualcomm Snapdragon X24 LTE, Wi-Fi 5 (a/b/g/n/ac) 2.4 + 5.0 GHz	UE

Table 4.1 Overview of devices used in the testbeds

4.1.1 Implementation - Local

For the development and implementation of CPF and COM into MPTCP, the testbed shown in Figure 4.3 played a central role. It has the benefit of a minimalistic setup with full control over all characteristics and parameters.

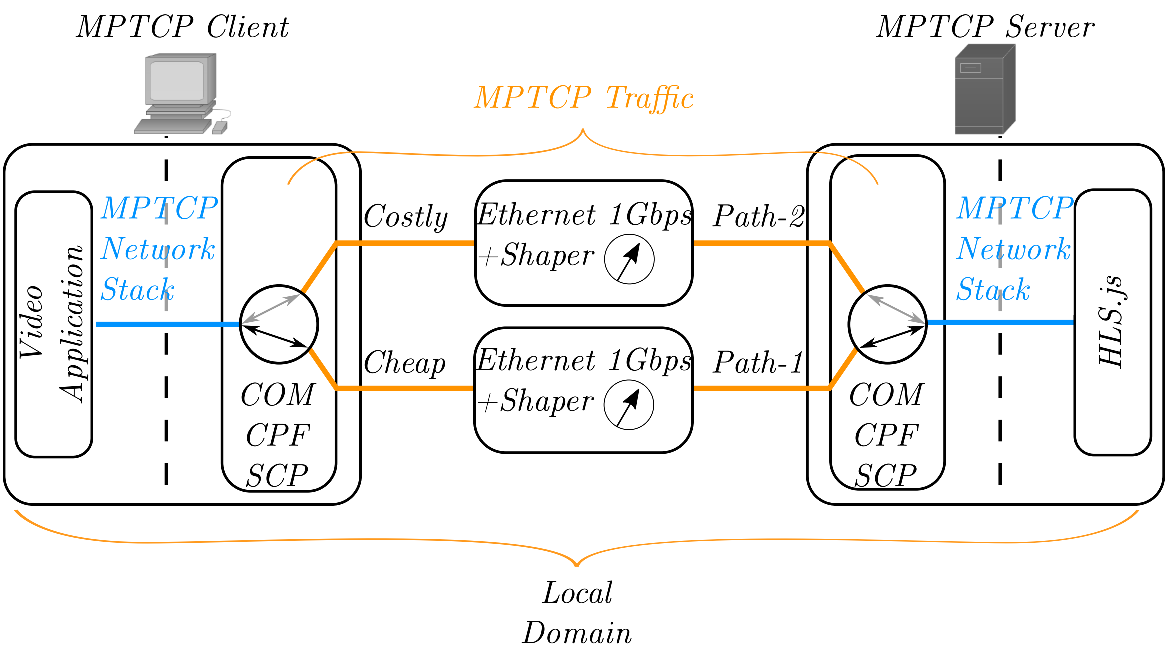


Fig. 4.3 Local testbed for CPF and COM initial design and implementation

Simply, the setup consists of two MPTCP enabled Linux PCs for testing with SCP, CPF and COM in a local domain. Two Ethernet connections, both with 1 Gbps form the backbone.

Traffic generation and consumption runs directly between the two entities using the MPTCP enabled network stack. Any service based on TCP like the traffic generator `iPerf` used with the `-t` flag automatically will establish a multipath connection across both Ethernet links with one subflow per link. The software `netem` for network emulation varies the characteristic of each link and for example allows to investigate the scheduling behavior under practical assumptions.

A variant of this testbed, not shown, uses a third Wi-Fi connection and was used to demonstrate the effect of CPF against the *minRTT* scheduler in subsection 3.2.4.

Since the local testbed fails to connect to Internet VoD services, *HLS.js*[4] was identified as alternative. As is also evident from the name of the application HLS is used as streaming protocol to provide chunks of a video.

If the video material is not available in the desired container format, resolution or codec, a conversion with for example `ffmpeg` can be executed:

```
ffmpeg -i Big_Buck_Bunny_4K.webm -vf scale=1920:1080
   Big_Buck_Bunny_1080.mp4
```

If the original video file is not using an HLS supported codec, the additional flag `-c:v libx264` can be added to above command to convert into the HLS supported H.264 codec. With `-crf`[5] also the quality can be modified if the default setting leads to undesired quality losses.

The individual chunks of the video which can be played out by *HLS.js* are finally generated with

```
ffmpeg -i Big_Buck_Bunny_1080.mp4 -codec: copy
   -start_number 0 -hls_time 10 -hls_list_size 0 -f
   hls Big_Buck_Bunny_1080.m3u8
```

and output the `m3u8` playlist file which can be requested by the video client, e.g. a browser, from the *HLS.js* server. The HLS specific flags[6] control the chunk size with `- hls_time` in seconds and were set to a size of 10 s, which is in the range of 4 s - 10 s, which has been found to be a typical time in the literature and in tests with VoD service providers.

[4]https://github.com/video-dev/hls.js
[5]https://ffmpeg.org/ffmpeg-all.html
[6]https://ffmpeg.org/ffmpeg-formats.html#hls-2

4.1.2 Implementation - Local with Internet

The here presented testbed extends the local testbed Figure 4.3 with the ability to integrate any Internet service but especially VoD providers like Youtube or Vimeo. Although this comes at the expense of full control it allows to verify the data from local testing in a more realistic environment. The key element which enables the connection of MPTCP based multipath transport to a regular TCP Internet services is a TCP Proxy.

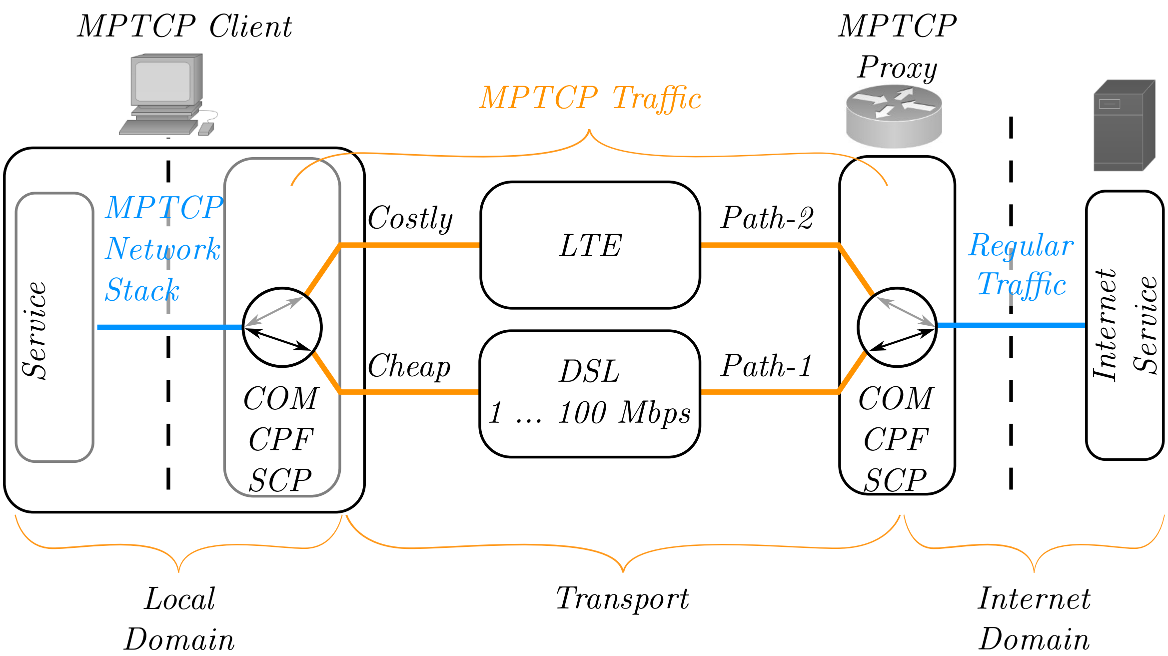


Fig. 4.4 Local testbed extended with a Proxy for the use of Internet services

While the transport in this enhanced testbed shown in Figure 4.4, shows commercial DSL and LTE accesses, another variant, not shown, still uses Ethernet links as shown in Figure 4.3, which has the advantage that the proxy remains in a controlled local environment. As this configuration did not help to achieve results in this work, it will not be explained further.

Following the logic of HA and ATSSS, the multipath transport is transparent to the Internet service and does not require any modification. Specifically with focus on the MPTCP, this needs a solution which converts between a regular TCP service in the Internet and the MPTCP transport. [131] proposes a TCP proxy solution, which terminates and re-opens TCP flows (also called split proxy). The principle is illustrated in Figure 4.5 which depicts a intermediary device – the MPTCP capable TCP proxy – (like ATSSS UE architecture in Figure 3.16) between a MPTCP client – left – and an unmodified one outer device unaware of multi-connectivity – right. An exemplary traffic flow is depicted from the left to the right, the bidirectional conversation is supported though. This simple approach triggers the conversion between TCP and MPTCP automatically by swapping the payload between the individual TCP and MPTCP session.

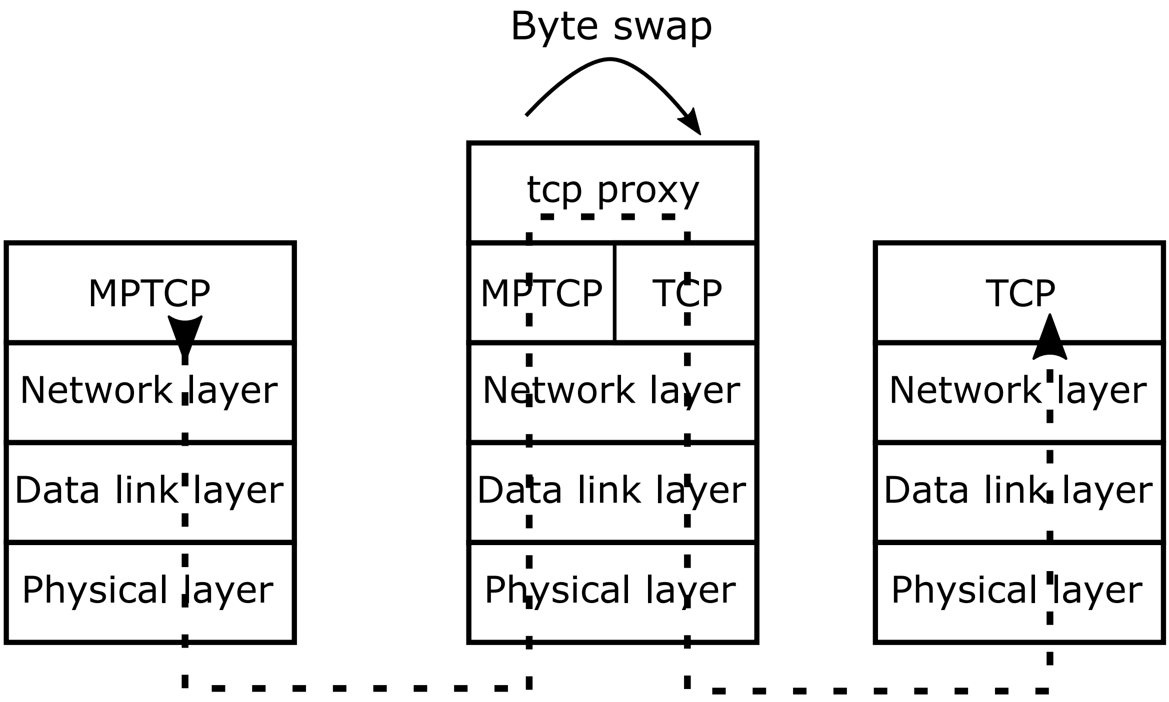


Fig. 4.5 Principle of transparent proxying for MPTCP ↔ TCP conversion

The connection establishment procedure shown in Figure 4.6 enables the bilateral communication flow between a multi-homed MPTCP client and a TCP server. The client starts the connection towards the MPTCP enabled proxy and also transmits the actual Server destination address $dest_{origin}$. This triggers the establishment of a second connection from the proxy to the Server, while the proxy is storing the source address src_{origin} (e.g., an IP address and a port) and the destination $dest_{origin}$ (e.g., an IP address and a port) as identifier to match both connections and to use it from now on to rewrite source and destination addresses between Client and Server depending on the traffic direction. After the establishment of the initial MPTCP flow between Client and Proxy, subsequent flows can be established – not shown – and the traffic scheduling method on Client and Proxy distributes the traffic between the initial and subsequent flow(s).

It should be noted that Figure 4.6 is a simplified method of explanation that may vary in some implementation detail and also does not take into account possible buffering of data between communication flows.

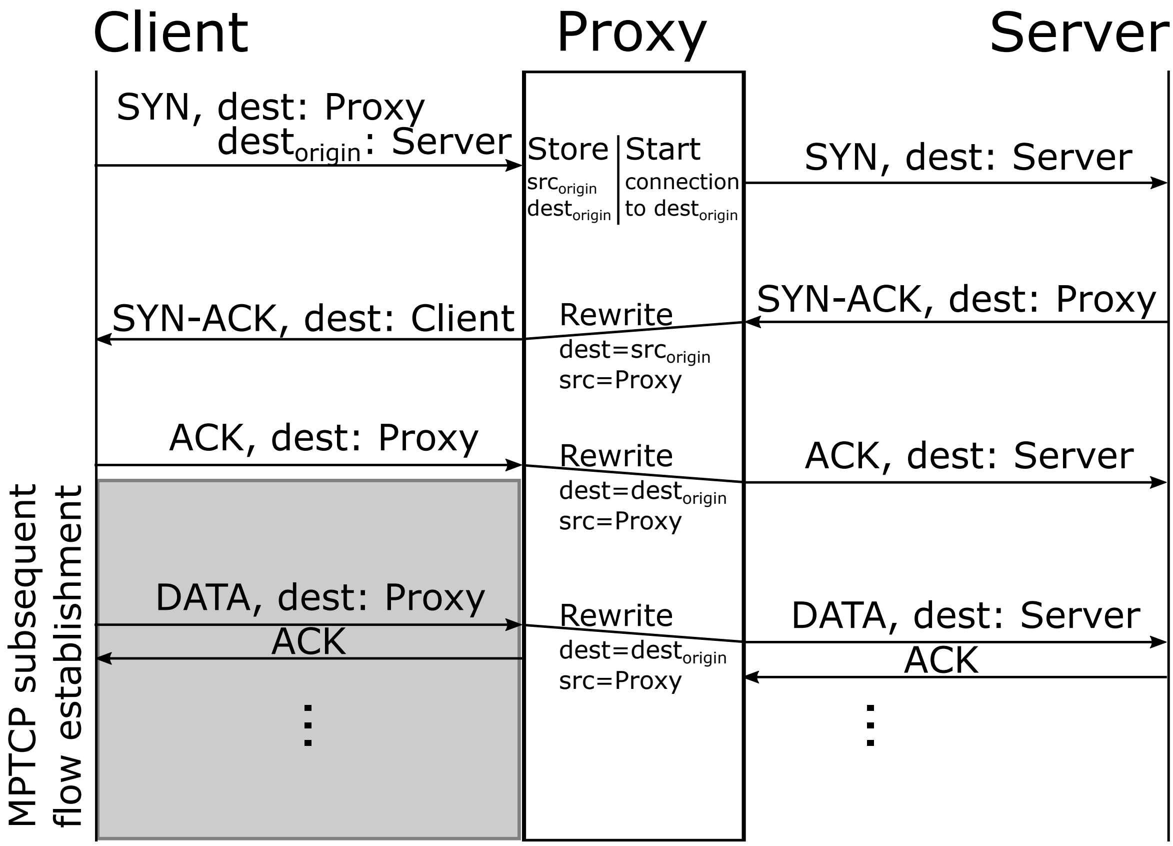


Fig. 4.6 Sequence diagram of a proxied connection establishement between a MPTCP client and a regular TCP service

This kind of proxy concept is also the selected method for the transparent use of the splitting function of ATSSS and is defined in [132]. In the end, this concept means that the payload is swapped between the receiver buffer of the TCP socket with incoming data and the send buffer of the forwarding TCP socket.

For the test environment, the open source TCP proxy *tcp-intercept*[7] was chosen, which follows exactly this concept even if it does not map the handshaking procedure of [132] one-to-one which is not crucial for the goal to assess the multipath scheduler.

tcp-intercept uses the Linux `TPROXY` functionality[8] which allows to transparently intercept incoming TCP or MPTCP connections with a non-local destination address and handle such traffic in the userspace to execute the payload swapping between the intercepted (MP-)TCP flow and a remote TCP flow towards the original destination.

[7] https://github.com/VRT-onderzoek-en-innovatie/tcp-intercept
[8] https://docs.kernel.org/networking/tproxy.html

During this work performance shortcomings of *tcp-intercept* were identified, solved and merged into the public code repository which are seen of relevance for commercial deployments.

One problem was the observation that if one of the connections between client and proxy or proxy and server was interrupted, this was not detected and the remaining connection remained forever unable to transport data. On the one hand, this blocked unnecessary resources and, on the other hand, did in case of doubt not cause the connection between client and server to be re-established via the proxy. A solution to overcome this was the usage of the Linux TCP socket option `SO_KEEPALIVE`[9] to monitor the permeability of the individual connections and destruct the TCP sockets if not.

Another observation was that the Nagle's algorithm feature of the Linux TCP stack caused high jitter between Client and Server which impacted the achievable throughput. This algorithm attempts to merge TCP data segments that are below an Maximum Segment Size (MSS) to make better use of transport capacity but introduces delays whenever this is applied. To disable this feature and force an immediate forwarding of data independent of its size, `TCP_NODELAY`[10] socket option was added.

4.1.3 Implementation - Hybrid Access & ATSSS

For the final evaluation of the COM scheduler in the target scenario of HA and ATSSS, a test environment is required that includes the essential units and functions needed to achieve the results that can be expected in the target scenario.

Since the ATSSS is a framework for providing transparent multipath splitting functionality for two scenarios, namely for the HA scenario as part of the BBF WWC architecture and for UE, both need to be considered. Another feature is the support of any TCP service but especially the one of VoD which is of particular interest for this work. This requires a testbed in which the probes required for the evaluation can be installed.

In relation to the HA scenario, Figure 4.7 is the resulting testbed that follows the essence of the ATSSS/WWC multipath model demonstrated in Figure 3.17. Compared to the local testbed with an interface to the Internet in subsection 4.1.2 the difference is the usage of a second Proxy. This is a necessary condition to be executed on the RG to keep transparency towards the connected consumer devices. This enhances the traffic flow between the endpoints as shown in Figure 4.8 with two times swapping the payload at the Proxies. A new TCP connection request between the endpoints will ultimately lead to three (MP-)TCP con-

[9] https://man7.org/linux/man-pages/man7/socket.7.html
[10] https://linux.die.net/man/7/tcp

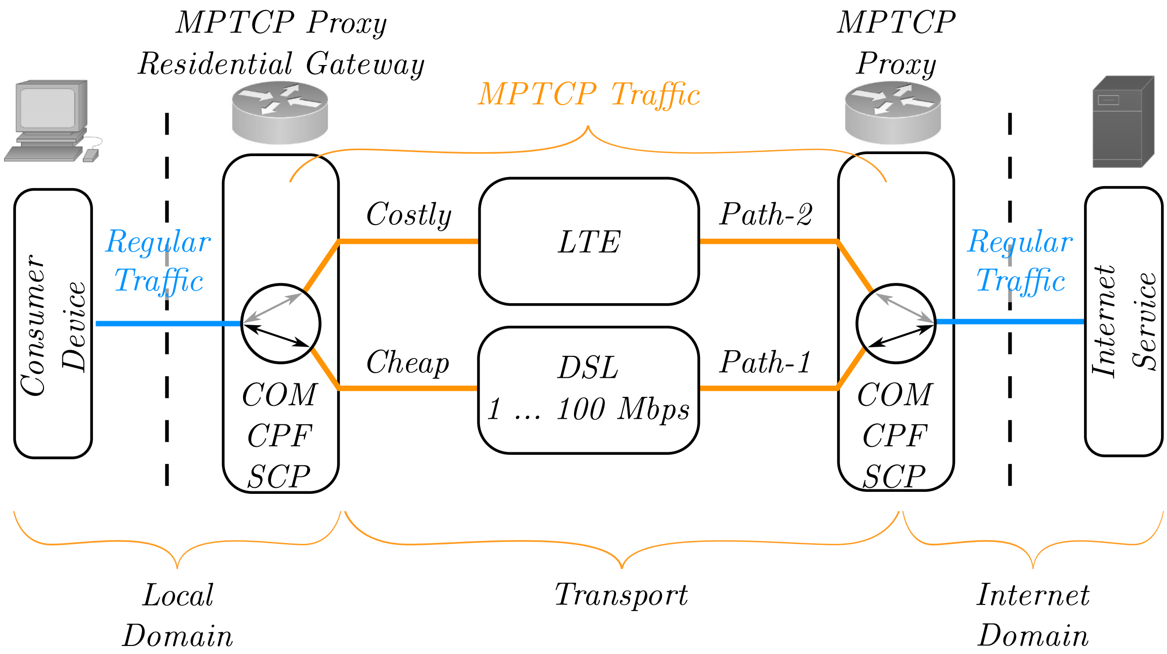


Fig. 4.7 Main test environment for COM evaluation following the Hybrid Access architecture with two Proxies for full transparency

nections between requesting endpoint and Proxy (regular TCP), Proxy and Proxy (MPTCP) and finally Proxy and serving endpoint (regular TCP).

The location of the device is spread across three different domains, namely the local, transport and Internet domains, with control spanning the consumer device, the RG and the Proxy that connects to the Internet. Uncontrolled is the commercial transport and the service in the Internet. To ensure minimal latency impact, the Proxy to the Internet was located in the same geographical area as the transport termination points ($\sim 200\,$km).

Apart from that, the conditions and descriptions of subsection 4.1.2 (local testbed with Internet access) are valid, as both testbeds Figure 4.4 and Figure 4.7 have a lot in common with the exception of a second proxy.

Even though in this work the demonstration of the conflict between cost and in particular VoD-transfer was done in an HA-environment, it is obvious that this will also affect the nomadic use case covered by ATSSS. Again following the ATSSS adapted multipath model in Figure 3.17, this results in a testbed shown in Figure 4.9.

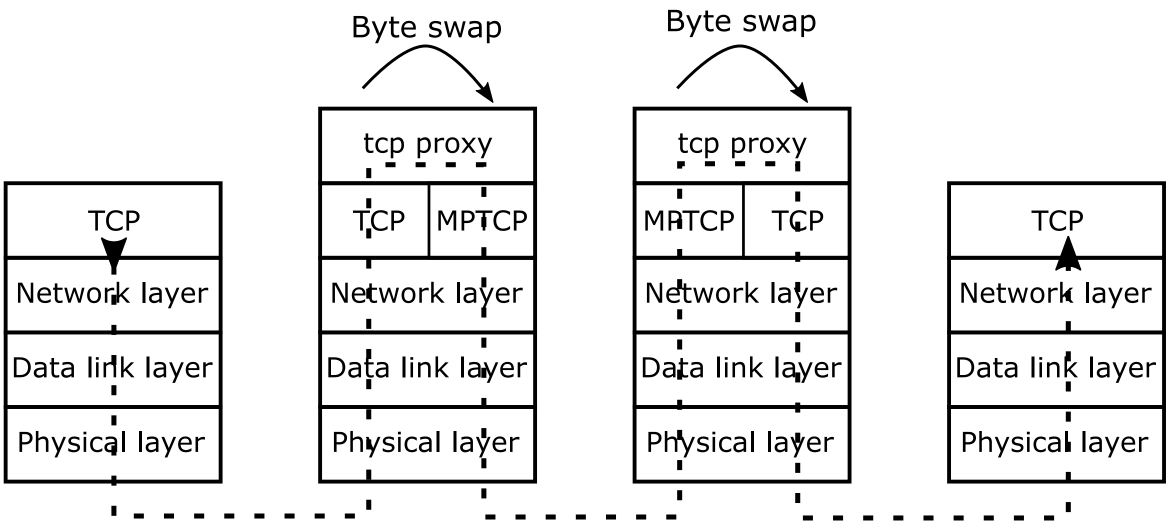


Fig. 4.8 Transparent TCP proxying in the Hybrid Access scenario according to the WWC architecture with two proxies

In some ways, the nomadic testbed is similar to the local testbed in Figure 4.4, with the MPTCP client replaced by a smartphone, but the differences and challenges are in the details. First, two wireless connections are used, with Wi-Fi replacing DSL access. Secondly, integrating with a smartphone with the aim of using it for a trial with mobile customers is challenging in many ways.

During the development of the testbed preliminary results showed comparable results with HA when the Wi-Fi was able to provide the same throughput and reliability as DSL. However, in the nomadic scenario if a smartphone moves around, the characteristic of the Wi-Fi will vary much because of the supported Wi-Fi standard, the signal quality and the connected backbone.

For the implementation in a smartphone which also finds acceptance by customers, a commercial smartphone with upscale features is best. Due to the requirement of an integrated MPTCP network stack with CPF and COM scheduler, the smartphone OS needed modification as no commercial smartphone with such functionality was available at the time of testing. These conditions limited the scope to smartphones which can be used with the Android Open Source code provided by the AOSP project[11]. Finally, this resulted in the selection of Google Pixel and Google Pixel 2 devices which were used for testing and trials in the years between 2016 and 2020.

The basis for Android is the Linux Kernel, which has simplified the adoption of the MPTCP stack used in this work to implement CPF and COM. Also the possibility to modify the User Interface was used to ensure multiple things:

[11]https://source.android.com

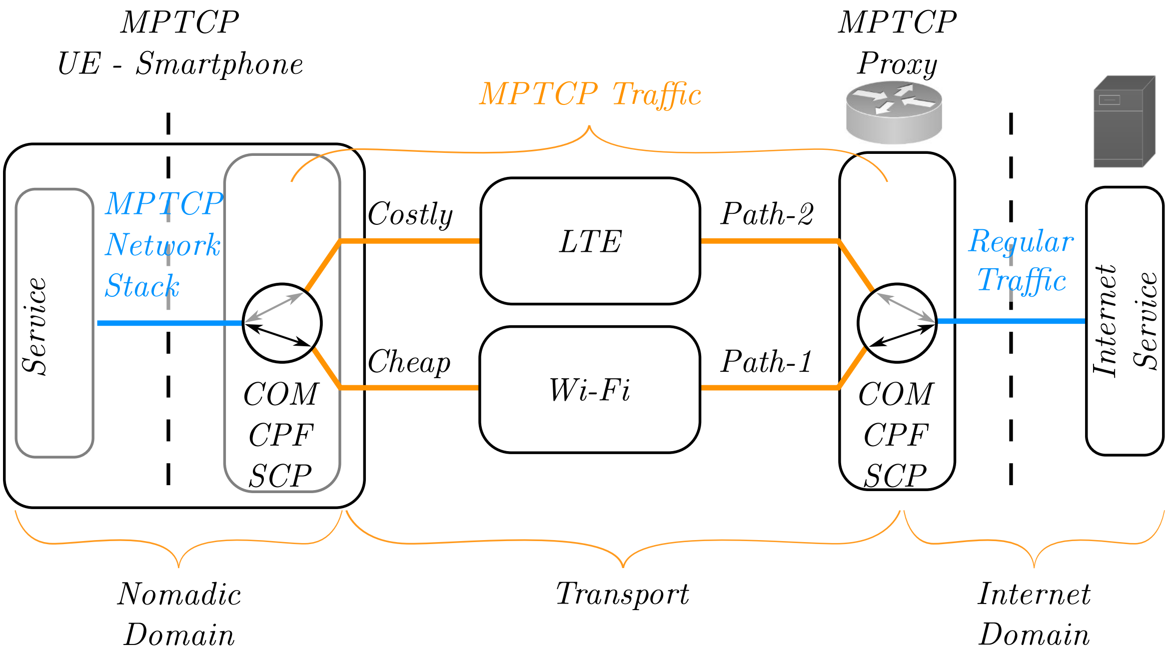


Fig. 4.9 COM evaluation framework for nomadic ATSSS scenario with UE

- Usage of MPTCP, selection/configuration of scheduler as well as congestion control for convenient re-configuration in test environment.
- No deactivation of the Wi-Fi interface to maximize multipath usage.
- Real-time throughput diagram in the network settings dialogue for visual verification of access related and combined throughput.
- API for remote configuration of settings for showcases and activation of different test phases during the field trial.

Another tool which has no impact to the scheduler performance but necessary especially for the field trial was the implementation of a UDP tunnel between smartphone and proxy per access. This solved three things that are testbed specific. First it helped to secure the connection between smartphone and Proxy. In a second aspect it avoided issues with MPTCP incompatible middleboxes [10, sec. 6] which let MPTCP fail. Lastly, it simplifies the separation of UE specific traffic for measurement and privacy.

In contrast, the network integrated ATSSS uses the security and UE identification methods of the 5G system. Also traffic over non-3GPP access is IPsec encapsulated and therefore does not expose MPTCP traffic to middleboxes.

A non-public Linux Kernel integrated tool was used to create, manage and execute the UDP tunnel which is a modified version of the related public DCCP encapsulation software[12] with full support of the maximum achievable throughput in the test and trial scenarios. The data overhead which comes with the usage of UDP tunneling because of additional IP and UDP header is considered negligible due to the minimal effect of $\sim 28\,\mathrm{B}$ (IPv6 transport was deactivated). In addition the effect of data overhead vanishes due to consistent overhead in all compared test scenarios.

Both testbeds together cover the case of ATSSS and HA and implement the relevant basis for the scheduler assessment with Internet services in chapter 5.

4.2 Comparative analysis of QoE and cost

The investigations of how to measure VoD QoE in section 2.5 using MOS, the reduction to the essentials of the MOS calculation in section 3.4 and finally the solution space depicted in Figure 4.1 provide the guidance needed to identify the right measures for the assessment of scheduling VoD traffic using COM.

The essence provides a QoE definition for VoD consumption which relies only on the playback experience. With $t_{initial}$, t_{stall} and the consumed video resolution r, the critical parameters which impacts this experience are determined. From a consumption point of view, those parameters should **at least result in the same or better experience than single path transmission under the prerequisite, that the single path is the cheap path and is a bottleneck.**

This leads to the fact, that the here evaluated QoE of VoD is mainly impacted by the available throughput, and that the other path capabilities plays a negligible role, as long as the BDP is covered by the $RecvBuffer_{VoD}$ and paths p_i exceeding $B_{mc} \cdot (L_i \pm J_i) > RecvBuffer_{VoD}$ are removed from the multipath transmission. Under this condition, QoE path characteristic dependency can be reduced to throughput B_p with a VoD QoE definition for the Single Cheap Path (SCP) scenario

$$QoE_{VoD,SCP} = \mathfrak{F}(B_{SCP}) \tag{4.1}$$

with B_{SCP} denoting the throughput capabilities of the single cheap path. Preferably, the single-path reflects the access path which is typically used for transmission without multi-path, for example, Wi-Fi in the smartphone or DSL in the home router. The focus on the

[12] https://github.com/telekom/tunprox/

throughput avoids more complex considerations arising from the non-linear behavior of other path characteristics like latency, jitter and loss which are typically covered by the $RecvBuffer_{VoD}$. In the multipath scenario, this definition holds true, however, the QoE is here dependent on the total bandwith B_{mc}, which is provided by the multiple paths in a multipath system.

$$QoE_{VoD,MP} = \mathfrak{F}(B_{mc}) \tag{4.2}$$

For getting an individual view on the QoE characteristics this work will compare the development of the start delay of a requested video ($t_{initial}$), the playback interruption (t_{stall}) and the received video resolution (r) **proportionally to a consistent video measurement time** $t_{measurement}$ with

$$t_{measurement} = const = t_{playback} + t_{initial} + t_{stall} \tag{4.3}$$

in the single path and the multipath scenario, while the latter is investigated using different scheduling algorithms. An outcome therefore is the determination of the QoE for the initial loading phase of the video and the interruption time with

$$QoE_{initial} = \frac{t_{initial}}{t_{measurement}} \tag{4.4}$$

$$QoE_{stall} = \frac{t_{stall}}{t_{measurement}} \tag{4.5}$$

For both values, the lower the values are with the limit value 0, the better the QoE becomes.

Finally, the QoE for the resolution is presented by the Euclidean norm of a vector according to the playback time spent resolutions as defined in [125]. With this and a weighting mechanism which grades the consumed resolutions it is ensured that the QoE value moves between 0 and 1. A higher value represents a better video resolution play backed over the measurement time.

$$QoE_{resolution} = \left\| \frac{\begin{bmatrix} t_{1920\,\text{px}} \cdot 10^{0} \\ t_{720\,\text{px}} \cdot 10^{-1} \\ t_{480\,\text{px}} \cdot 10^{-2} \\ t_{360\,\text{px}} \cdot 10^{-3} \\ t_{240\,\text{px}} \cdot 10^{-4} \end{bmatrix}}{t_{measurement}} \right\| \tag{4.6}$$

The best QoE that can be achieved under these conditions over all QoE values in Equation 4.4 - 4.6 is a $QoE_{resolution} = 1$. This automatically leads to $QoE_{initial} = QoE_{stall} = 0$ and means an uninterrupted consumption of a VoD stream with highest resolution over the entire measurement time. Overall the sum of the QoE values can not exceed 1 in the upper bound and 0 in the lower bound.

In a comparative approach the above QoE values can be used to calculate a gain γ and evaluate the QoE benefit across single-path (*SCP*) and multi-path (*MP*) or between different multi-path scheduling algorithms $A1$ and $A2$, for the *initial* gain

$$\gamma_{initial} = \begin{cases} 1, & \text{if } QoE_{initial,SCP} = 0 \ \& \ QoE_{initial,MP} = 0 \\ max, & \text{if } QoE_{initial,SCP} \neq 0 \ \& QoE_{initial,MP} = 0 \\ \dfrac{QoE_{intial,SCP}}{QoE_{initial,MP}}, & \text{otherwise} \end{cases}$$
$$or \tag{4.7}$$
$$\gamma_{initial} = \begin{cases} 1, & \text{if } QoE_{initial,MP,A1} = 0 \ \& \ QoE_{initial,MP,A2} = 0 \\ max, & \text{if } QoE_{initial,MP,A1} \neq 0 \ \& \ QoE_{initial,MP,A2} = 0 \\ \dfrac{QoE_{intial,MP,A1}}{QoE_{initial,MP,A2}}, & \text{otherwise} \end{cases}$$

and corresponding for the *stall* and *resolution* gain with

$$\gamma_{stall} = \begin{cases} 1, & \text{if } QoE_{initial,SCP} = 0 \ \& \ QoE_{initial,MP} = 0 \\ max, & \text{if } QoE_{initial,SCP} \neq 0 \ \& \ QoE_{stall,MP} = 0 \\ \dfrac{QoE_{stall,SCP}}{QoE_{stall,MP}}, & \text{otherwise} \end{cases}$$
$$or \tag{4.8}$$
$$\gamma_{stall} = \begin{cases} 1, & \text{if } QoE_{initial,MP,A1} = 0 \ \& \ QoE_{initial,MP,A2} = 0 \\ max, & \text{if } QoE_{initial,MP,A1} \neq 0 \ \& \ QoE_{stall,MP,A2} = 0 \\ \dfrac{QoE_{stall,MP,A1}}{QoE_{stall,MP,A2}}, & \text{otherwise} \end{cases}$$

$$\gamma_{resolution} = \begin{cases} 1, & \text{if } QoE_{initial,MP} = 0 \ \& \ QoE_{initial,SCP} = 0 \\ max, & \text{if } QoE_{initial,MP} \neq 0 \ \& \ QoE_{resolution,SCP} = 0 \\ \dfrac{QoE_{resolution,MP}}{QoE_{resolution,SCP}}, & \text{otherwise} \end{cases}$$
$$or \tag{4.9}$$
$$\gamma_{resolution} = \begin{cases} 1, & \text{if } QoE_{initial,MP,A1} = 0 \ \& \ QoE_{initial,MP,A2} = 0 \\ max, & \text{if } QoE_{initial,MP,A1} \neq 0 \ \& \ QoE_{resolution,MP,A2} = 0 \\ \dfrac{QoE_{resolution,MP,A1}}{QoE_{resolution,MP,A2}}, & \text{otherwise} \end{cases}$$

This analysis of γ will give a clear evidence of the effect and impact between multi-path scheduling algorithms and even over single-path transmission for VoD QoE. According to the objective of this work, the QoE development has to be brought in relation to the path cost. The path cost factor θ is defined as the load of the costly path compared to the total load generated by the measured service (e.g., VoD). For this the definition C_{mc} in Equation 3.4 provides the basis under the prerequisite that the cost of the single path C_{SCP} which also corresponds to the cheap path, as outlined above, is set to: $C_{SCP} = 0$. This eliminates the cheaper path from the cost calculation in Equation 3.4 and results in a θ of:

$$\begin{aligned} &\theta_{SCP} = 0, \quad \text{if cheap singlepath is measured} \\ &or \\ &\theta_{MP} = C_{mc}, \quad \text{if multipath is measured} \end{aligned} \tag{4.10}$$

with a $\Delta\theta$ to compare the costs incurred between singlepath and multipath or between two multipath scenarios:

$$\begin{aligned} &\Delta\theta = \theta_{SCP} - \theta_{MP} \\ &or \\ &\Delta\theta = min(\theta_{MP,A1}, \theta_{MP,A2}) - max(\theta_{MP,A1}, \theta_{MP,A2}) \end{aligned} \tag{4.11}$$

In the final analysis of multi-path scheduling algorithms both γ and $\Delta\theta$ need to be evaluated and will be interpreted as follows.

$$\begin{aligned} \gamma > 1, &\quad \text{better QoE of MP over SCP or MP,A2 over MP,A1} \\ \gamma = 1, &\quad \text{no QoE change} \\ \gamma < 1, &\quad \text{worse QoE of MP over SCP or MP,A2 over MP,A1} \end{aligned}$$

$$\begin{aligned} \Delta\theta = 0, &\quad \text{no additional cost by MP or MP,A2} \\ \Delta\theta < 0, &\quad \text{additional cost by MP or } max(\theta_{MP,A1}, \theta_{MP,A2}) \end{aligned}$$

The Equation 4.4 - 4.11 will be used in chapter 5 to evaluate the performance of COM.

4.3 Data collection

After defining the testbed (section 4.1) and the relevant parameters for assessing costs and QoE impacts (section 4.2), this section serves to identify what needs to be measured and how this should be done.

The most relevant testbeds for the purpose of the assessment in an architecture resembling ATSSS for either the HA scenario (Figure 4.7) or the UE scenario (Figure 4.9) allow the installation of probes in different locations. This possibility extends through the consuming device, the RG and the proxy towards the Internet but exclude probes in the commercial transport and Internet service.

To measure the path cost factor (Equation 4.10), it finally needs the monitoring of the utilized bandwidth of the individual accesses according to Equation 3.4. This can be measured by counting the volume sent across the individual access path in the scheduling entity (Proxy, RG or UE) related to the measurement target. In the simplest case the byte counter of the cellular and fixed Wi-Fi interface are monitored using the Linux information provided as summary for all network devices in `/proc/net/dev` or using the device individual statistics under `/sys/class/net/*devname*/statistics`. In the case that a distortion of the measurement by cross traffic threatened, e.g., in the case of multiple field trial UEs connected to the proxy, traffic accounting using `iptables` or the use of a network analyzer like `tcpdump` served as solution.

More effort is required to determine the QoE impact which is evaluated in this work (chapter 5) with focus on:

- Video-on-Demand (VoD)
- Website requests
- Overall traffic mix of smartphone users

In the VoD case this requires the collection of $t_{initial}$, t_{stall} and the playback time $t_{playback}$ distinguished by the consumed video resolution.

This is solved at the client side with a scripted environment using Chromium browser to request automatically a specified Youtube video over TCP. To avoid effects from arbitrary advertisements during start or run of the video a Chromium plugin *uBlock Plus Adblocker* was used. While this helps to make test iterations reproducible, it also uses the Youtube embedded player to load only the video and no surrounding information which are usually displayed on the Youtube website. Along with the embedded player the Youtube Iframe API[13] provides access to the information required to determine the QoE parameters identified above and can be accessed for example through the logging feature of Chromium. Finally, the Chromium plugin *Youtube Auto HD + FPS* allows to specify for a certain video resolution and gets rid of Youtube's by default enabled Adaptive Bit Rate (ABR) which lowers the resolution if throughput is not sufficient. This helped to test the case of static video resolution playout and the one of ABR.

Another typical usage scenario is the request of a website. This is interesting in terms of the initial loading time of a website when the parameter COM $T_{StartDelay}$ is used to delay the expensive path usage right after the connection is established.

Like in the VoD case, the same Chromium browser is used to reproducible measure the time of loading a website with all its content. Also here the advertisement blocker is used to avoid arbitrary effects and Adobe Flash was deactivated. When Chromium is started with `-enable-logging=stderr`, all relevant information is output to the standard OS error stream, from where it can be retrieved for automatic further processing by scripts to determine the time between the website request and its full delivery.

[13] https://developers.google.com/youtube/iframe_api_reference

In the last case, it was necessary to gain impressions about the behavior in interaction with any kind of traffic. Since this is solved by a smartphone implementation of COM provided to the customers of a mobile operator, it was difficult to actually measure the QoE, since this is only possible if the service under test is known. However, a clever segmentation of the trial into different measurement phases accompanied by a customer survey made it possible to achieve results here as well.

Chapter 5

Results and Analysis

In the methodology chapter 4 a clear mandate is formulated to investigate and compare COM over CPF and Single Cheap Path (SCP) for determining cost and QoE gains in the ATSSS scenario for HA and UE multipath transport. It is further specified to focus on the interaction with VoD as the main optimization goal of this work. On the other hand it is also stated that a certain amount of evidence should be provided that COM can replace CPF without limitations in respect to other type of traffic.

This chapter aims to present the results of the testing described in detail above and also to provide some background information on the measurements carried out during the initial design and implementation phase of COM that motivated the continuation of this idea.

When COM was initially designed in subsection 3.5.3, $t_{GAPthresh}$ for detecting burstiness and T_{Delay} for preventing temporary access to high cost path formed the parameter set. In section 5.1 it is shown that reasonable results were achieved in a local testbed showing that the high cost path consumption significantly reduces with COM. However, the testing also revealed some shortcoming which let COM fail to reliably detect burstiness and lead to a further enhancement of COM.

In the course of further testing, the focus is on the enhanced COM logic (subsection 3.5.4) with new $t_{notraffic}$ and V_{max} parameter and COM's interplay with the Internet VoD traffic, and this chapter includes a range of testing results. Firstly, section 5.2 demonstrates results of extensive tests with controlled VoD traffic over the Internet, used to determine the initial COM parameter set. This reduces the number of measurement variables to T_{Delay} and DSL throughput. This is then further used for testing with Youtube traffic in section 5.3 to measure the cost and QoE when VoD with static video resolution is scheduled. In a further step in section 5.4 this measurement is enhanced towards VoD with dynamic video resolution, which needs a more comprehensive QoE consideration.

With the confidence from the promising results of the previous tests, other types of non-VoD traffic will also be investigated in section 5.5 and section 5.6. The first type of tests are executed with regard to the impact of $T_{StartDelay}$ on the QoE and cost when websites are requested which is a typical workload in Internet communication. The second type of test corresponds to an trial with the aim of understanding the impact of COM on a typical traffic mix (with VoD and without VoD) consumed by users in daily Internet use.

Finally, section 5.7 summarises and analyses the measurement results.

All tests – not trials – presented throughout this chapter were conducted once within the testbed using the DSL and LTE access (or equivalent shaped local links) as described in chapter 4. Collecting only one test sample is not a limitation, as the evidence of validity is drawn from the trend of results. To avoid the test system being biased by cached values from previous tests, the system was reset after each run of a sample.

Some remarks on interpreting the results throughout this chapter:

- Line graphs: The lower the values the better.
- Bar charts: The more blue the better
- General: $T_{Delay} = 0$ means CPF principle, $T_{Delay} = inf$ means single usage of low-cost path

5.1 Initial testing of COM

The initial tests of COM accompanied the first implementation phase (subsection 3.5.3) and served to understand whether the basic idea of COM is going in the right direction and whether the performance is similar to that of CPF and the MPTCP *default* scheduler. It is also important to understand that this initial phase does not include the parameters introduced in the final implementation of COM, namely $t_{notraffic}$ and V_{max}. This was a result of the initial testing discussed here and was subsequently introduced into subsection 3.5.4 as a refinement of the COM algorithm. In this test phase, the local testbeds with (subsection 4.1.2) and without internet connection (subsection 4.1.1) were used.

A first comparison of the scheduling behaviour between CPF and the MPTCP *default* scheduler is presented in subsection 3.2.4 with a clear cost difference in favour of CPF.

What is not discussed there and is also not visible in the results was the performance test that was part of this step. Here, the local testbed with the MPTCP was run at the limit of the client and server capabilities, not those of the transport. In this case, the CPU limited the maximum achievable throughput in the system, but both schedulers achieved the same maximum throughput.

Similar tests were carried out to determine the maximum achievable throughput of the *COM* scheduler, which ended with the same result. No measurable computational impact is caused by taking timestamps, comparing them, setting and verifying T_{Delay}. Parts of this time-related processing are handled in TCP anyway and can easily be reused by COM. This suggests that, at least in a system where MPTCP and scheduling are run on one CPU, there is no performance penalty in choosing COM. It can also be concluded that this finding feeds directly into the design principle of simplicity formulated in subsection 3.5.2.

When QoE is considered for Video-on-Demand (VoD) services, the good quality of experience is indicated by a smooth playback of video. Whenever the video playback is disrupted (frozen) a reduction of QoE is noticed. It is obvious that the focus on consumer QoE includes the network related transmission characteristics like loss, latency and jitter. If at least one of these fails to ensure the expected levels, the video playback starts stuttering. Furthermore, to validate the impact of the *COM* algorithm, its performance is compared to: 1) single path transmission over the cheaper path and 2) the *CPF* algorithm, both according to Figure 4.1. It should be noted that focusing on freezes as the only indicator at this early stage of testing does not correspond to the results of the extended set of QoE parameters in section 4.2, but this was assessed as a good enough indicator to decide to continue the COM development, but on the other hand to keep the effort manageable. The same applies to the selection of test parameters presented in this section, which do not aim to identify the best parameter set for a wide range of access throughput scenarios. This will be done in the later sections of this chapter.

Figure 5.1 compares the amount of cost optimization (left y-axis) while monitoring the QoE (right y-axis) in terms of occurrences of video freezes. Both are depicted as function of T_{Delay}, which indicates the time access to the more expensive path is denied after traffic burst is identified based on Equation 3.8. T_{Delay} is investigated in an interval $[0\,\text{s}; 7\,\text{s}]$ whereas $0\,\text{s}$ means effectively *CPF* (no optimization) and $7\,\text{s}$ points towards single path behaviour. In accordance with the local testbed Figure 4.3, the MPTCP client/server setup is deployed, with the server providing the same 1080p/60fps video as in footnote 9, however locally based on hls.js. This HLS javascript library is also used to count the number of freezes. The

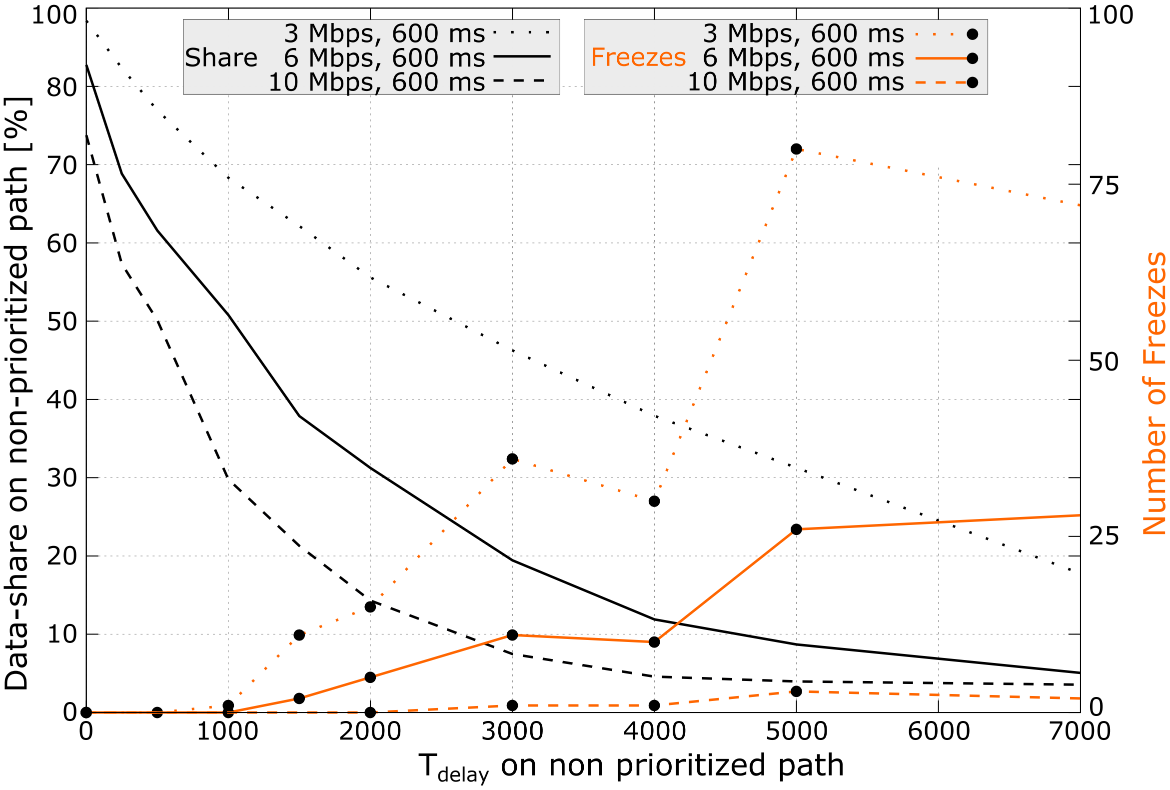


Fig. 5.1 First COM - Costly share of different C_{cheap} over T_{Delay} and number of playback freezes in a local testbed with $T_{GAPthreshold} = 600\,\text{ms}$

measurement period corresponds to the video length of approximately 10 min and an average throughput of 8 Mbps. The prioritized Ethernet link, denoted as C_{cheap} is shaped with means of `tc` in a bridging device to 3, 6 and 10 Mbps, while $T_{GAPthresh}$ is fixed to 600 ms, a size that was found to be good by trial and error. The secondary, non-prioritized path, is left unchanged at 1 Gbps, large enough to not represent a bottleneck in case C_{cheap} has limitations. In hls.js, the video segment size was set to 4 s and the buffer size on client side was set to two segments effectively lead to a pre-load of up to 8 s of video material. Measurements were only conducted once due to the local controlled nature of the testbed with a high expected reproducibility. Starting with the non-optimization ($T_{Delay} = 0$) case, one can see for all tested C_{cheap} values a very high consumption of the de-prioritized resources roughly between 70 % and 90 %. With an increasing T_{Delay}, this share can be significantly reduced by multiple decades and that even with already very small T_{Delay}. At the same time at around 1 s the total number of video freezes starts to grow. This is expected as the prioritized bottleneck path is prevented to aggregate a secondary path for some time. Nonetheless, comparing this with the freezes at high T_{Delay}, which seem to be close to single path transmission, one can identify a wide period in between, which indicates both, a significant reduction of costly resource usage and an acceptable level of QoE with a clear benefit when no multipath is applied. In terms of the Figure 4.1 operation point, a significant shift from the right to the left (costly resource usage) with a minor shift from the top to the bottom (QoE) can be identified. This shows the clear potential of COM even at this early stage. With the T_{Delay} parameter of COM, the costs and, as a preliminary QoE indicator, the freezes can be balanced in the present test scenario in order to achieve a cost reduction while maintaining a certain QoE level. This is a first encouraging result with regard to the objectives of the work.

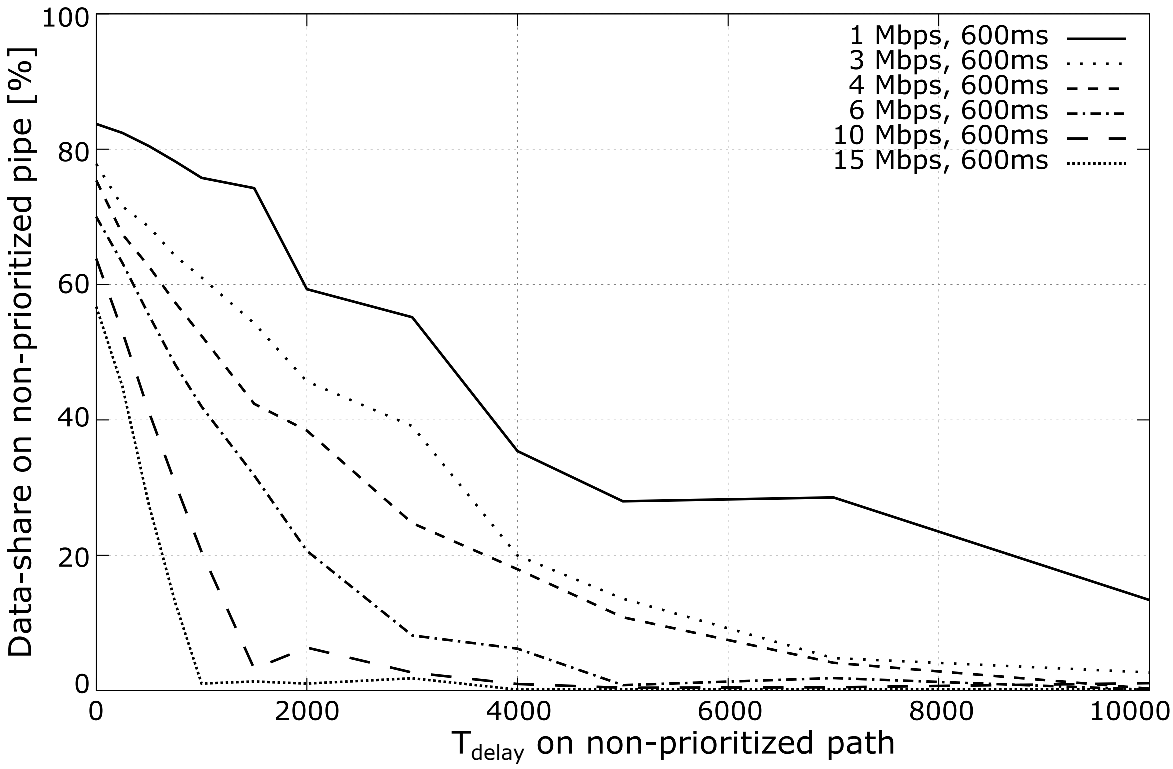


Fig. 5.2 First COM - Costly share of different C_{cheap} over T_{Delay} with Youtube and $T_{GAPthreshold} = 600\,\text{ms}$

While in Figure 5.1 *COM* was verified in a local testbed, Figure 5.2 uses the Proxy offered Internet connectivity from the online testbed Figure 4.4 to request a YouTube video[1] with auto-resolution limited to 1080p – static resolution. Similar to the local testbed a significant offload of the costly resource is gained. Even with small T_{Delay} values the LTE share can be brought down to almost zero for a $C_{cheap} \geq 6\,\text{Mbps}$. An issue which appeared occasionally with YouTube during this measurement was, that during the transmission of portions of the videos, control messages were exchanged which render the burstiness detection based on

[1]https://youtube.com/watch?v=aqz-KE-bpKQ

Equation 3.8 useless.

Although these initial results do not yet give a comprehensive overview of the optimisation potential of COM, and a major problem was also found in Youtube tests that occasionally breaks T_{GAP} detection, the first impression however confirms a value of COM for both cost reduction and QoE consistency within a certain range.

Because nothing in the limited tests indicated that this is not transferable to the extended scenarios of ATSSS, COM has been extended to address the problem with real services in subsection 3.5.4, which serves as the basis for the comprehensive testing in the following sections.

5.2 Determination of COM initial parameter set

For the purpose of testing COM across different Internet use cases, it is necessary to derive an initial parameter set. This means basically to evaluate if $t_{GAPthresh} = 600\,\text{ms}$ as selected in section 5.1 continues to be appropriate and moreover to determine reasonable values for $t_{notraffic}$ and V_{max} which form the enhanced COM parameter set as specified in subsection 3.5.4.

In Figure 5.3, Figure 5.4, Figure 5.5 and Figure 5.6 the impact of $t_{GAPthresh}$ across different DSL throughput of 1 Mbps, 2 Mbps, 6 and 16 Mbps is investigated within a range of T_{Delay} from 0 s-10 s in the Internet connected testbed (subsection 4.1.2). As can be seen in the figures, this is a reasonable range of DSL throughputs, with no relevant deviations at and beyond 16 Mbps to infer the parameter set that is the target here. It must be noted that $T_{Delay} = 0\,\text{s}$ corresponds to the CPF scheduling principle - non effective COM - while the larger T_{Delay} becomes, the traffic is finally scheduled on DSL only. This is in particular because an initial delay $T_{StartDelay} = T_{Delay}$ after connection establishment is applied without needing a $t_{GAPthresh}$ calculation. Based on experiments and observations, a static $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$ was configured to keep the focus on the change of LTE consumption and number of freezes at $t_{GAPthresh} = \{200, 400, 800\}\,\text{ms}$. In the case T_{Delay} increases, this reduces the achievable throughput B_{mc} in the multipath system. The higher T_{Delay} is configured, the more B_{mc} is reduced and causes an increasing number of freezes due to insufficient transmission capacity to fill the clients' playout buffer. Even if the number of freezes does not give all details of the QoE measurement, e.g. the length of freezes, it is

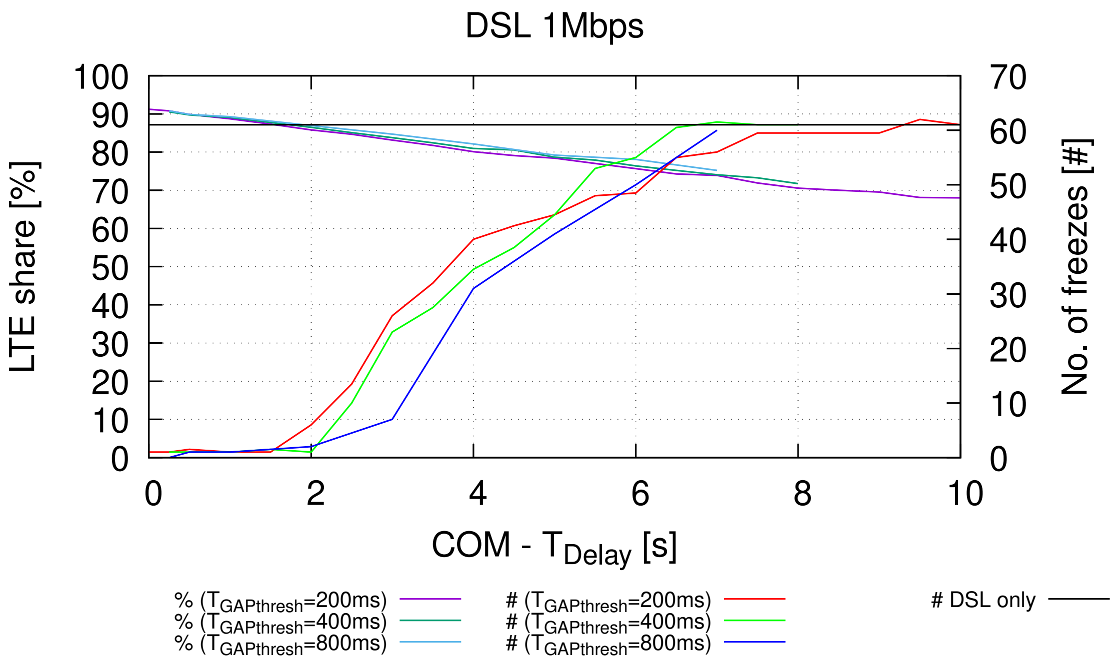


Fig. 5.3 LTE share and video freezes for variable $t_{GAPthresh}$ at 1 Mbps DSL rate, $t_{notraffic} =$ 50 ms and $V_{max} = 100$ pkts

sufficiently accurate to get an indication. All four results streaming a 1080p video[2] show LTE consumption which is decreasing with the increasing T_{Delay}, while freezes are appearing later in time, with this point increasing with the increasing DSL throughput. This is an expected result from the initial testing in section 5.1, however the focus is still not on the absolute gain, but rather on the relative trend. This development shows that the different values of $t_{GAPthresh}$ have minimal impact on the LTE share, as lines are close together - at least for 200 ms and 400 ms. Especially at higher T_{Delay} a $t_{GAPthresh}$ of 600 ms has smaller disadvantages. The opposite can be found when analyzing the number of freezes, as a result of greater use of the LTE capacity. Overall, a $t_{GAPthresh}$ between 200 ms and 400 ms does not result in performance differences, while between 400 ms and 800 ms a trade-off between the video freezes and the LTE capacity consumption exists. **As a reasonable trade-off further measurements are continued with** $t_{GAPthresh} = 600ms$**.**

Compared to the LTE shares from the initial testing as well as the results shown later, the gradient is not as pronounced. This is due to the fact that in this test phase no dedicated DSL connections were available for each bandwidth, but only one with 100 Mbps. `tc` was used to limit the DSL bandwidths, which does not correspond to the exact behavior. However, this is

[2] Big Buck Bunny 1080p video consumed with hls.js demo server

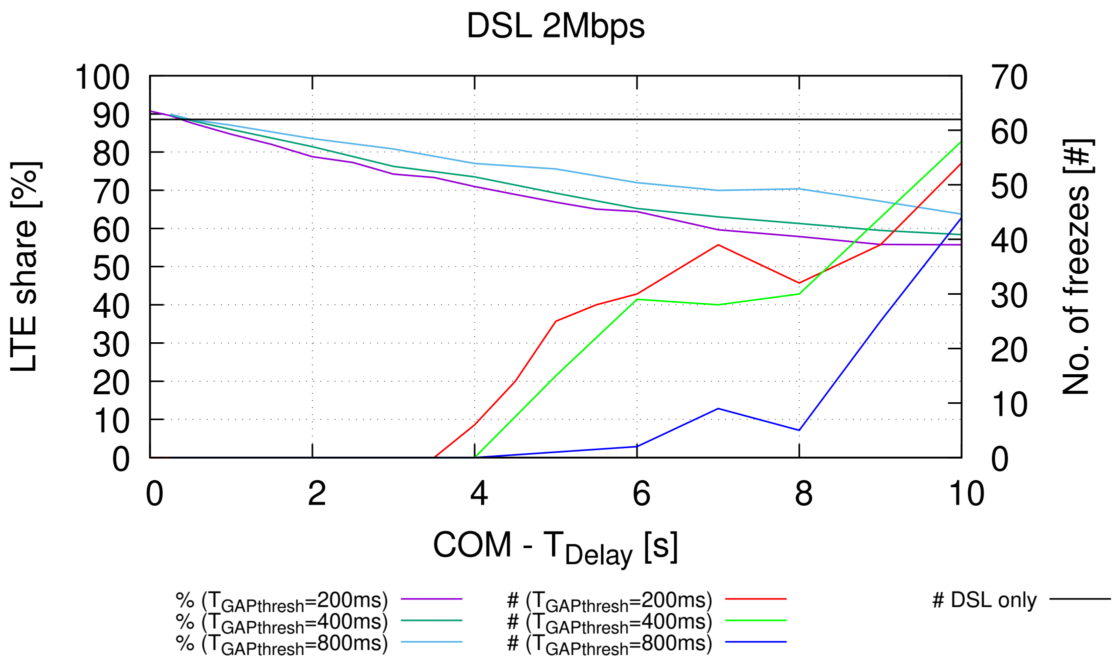


Fig. 5.4 LTE share and video freezes for variable $t_{GAPthresh}$ at 2 Mbps DSL rate, $t_{notraffic} = 50$ ms and $V_{max} = 100$ pkts

irrelevant for the result to be achieved, since the trend is decisive and this is correct. This also applies to the number of freezes, which also detected the initial delay, for example, and thus also indicate a freeze for the 16 Mbps result. In the later tests, more refined methods were used to determine QoE, which exclude such inconsistencies.

Tests – not shown – at higher DSL throughput, e.g., 50 Mbps, as well as running the measurement series with 720p video streaming confirmed the analysis above showing the same trend of results in respect to $t_{GAPthresh}$.

For the evaluation of $t_{GAPthresh}$, static values of $t_{notraffic} = 50$ ms and $V_{max} = 100$ pkts were used as this showed good results during some earlier experiments. As part of the next step to determine an initial COM parameter set, $t_{GAPthresh} = 600$ ms is used as static value and $t_{notraffic}$ and V_{max} are varied for the 2 Mbps DSL access. The latter results from the fact that the 2 Mbps evaluation in Figure 5.4 shows the largest variance compared to the other presented DSL throughputs in Figure 5.3, Figure 5.5 and Figure 5.6. As the upfront experiments with $t_{notraffic} < 50$ ms did not provide any positive results, the exploration range is set to $t_{notraffic} = \{50, 100, 200\}$ ms using 50 ms as starting point and $V_{max} = \{50, 100, 200\}$ pkts with 100 pkts as medium value. Again the impact on the LTE share and the number of freezes is monitored in Figure 5.7a and Figure 5.7b. Also here, a higher LTE share keeps the freezes

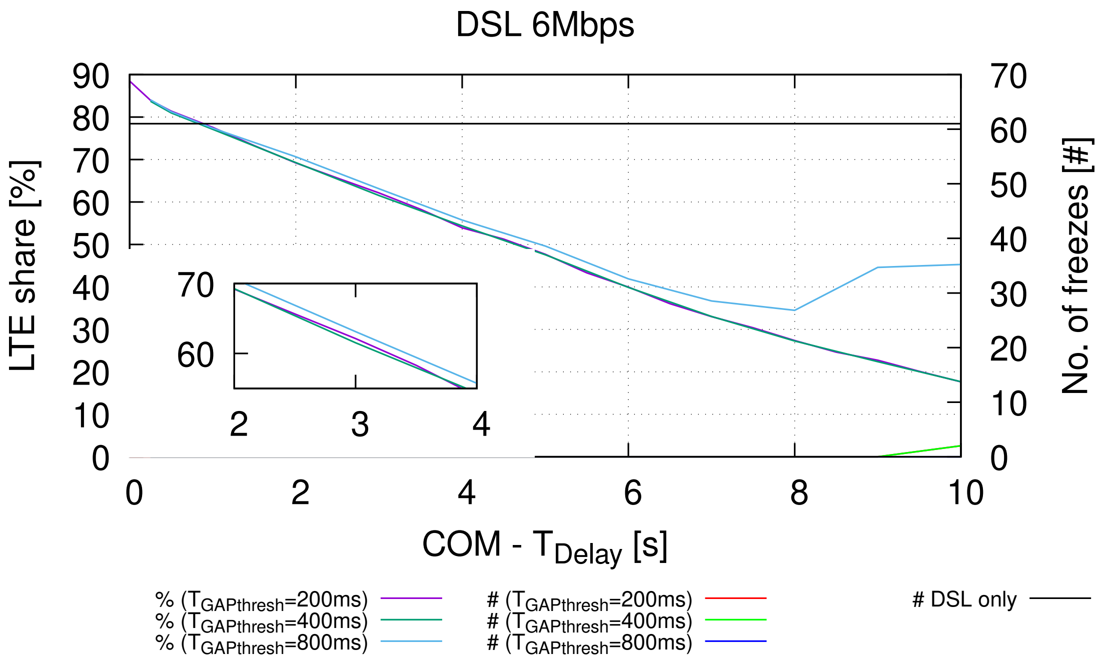


Fig. 5.5 LTE share and video freezes for variable $t_{GAPthresh}$ at 6 Mbps DSL rate, $t_{notraffic}$ = 50 ms and V_{max} = 100 pkts

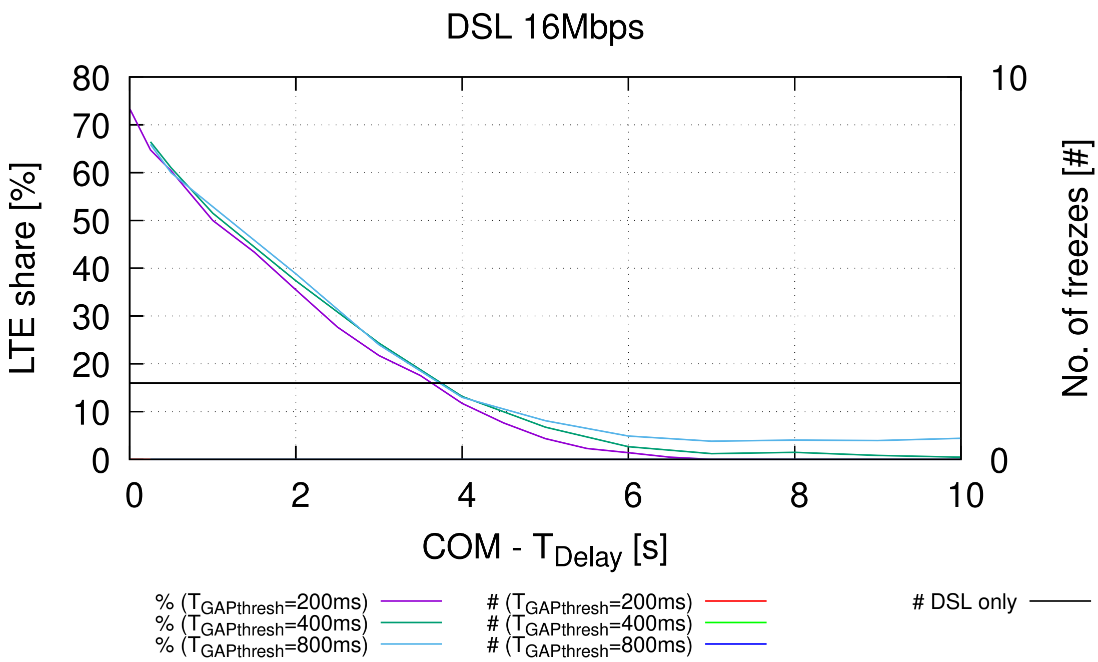
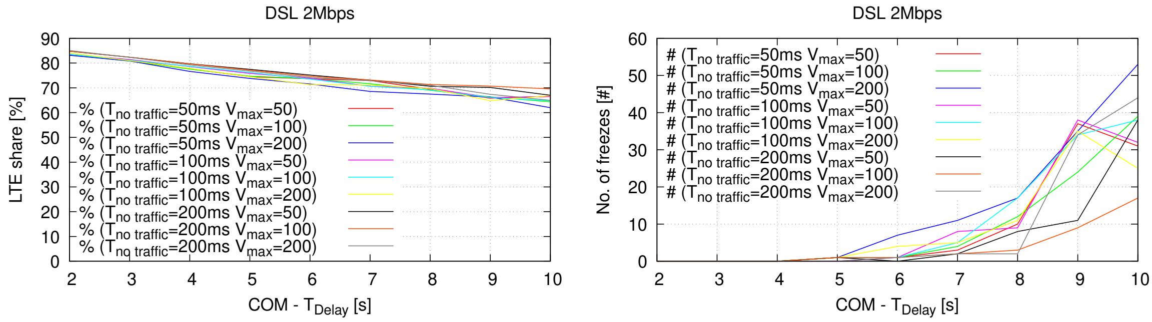


Fig. 5.6 LTE share and video freezes for variable $t_{GAPthresh}$ at 16 Mbps DSL rate, $t_{notraffic}$ = 50 ms and V_{max} = 100 pkts

low and vice versa. It is therefore advisable to continue with the results which provides the best compromise which is the set of $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$.



(a) LTE share (b) Video freezes

Fig. 5.7 LTE share and video freezes for variable $t_{notraffic}$ and V_{max} at 2 Mbps DSL rate and $t_{GAPthresh} = 600\,\text{ms}$

Overall, the conducted tests have shown that cost and QoE are mainly impacted at lower throughputs of the low cost path, when the identified parameters are varied. The multiple degree of freedoms of the COM scheduler are reduced with the identified parameter set for $t_{GAPthresh}$, $t_{notraffic}$ and V_{max}, to solely T_{Delay}. With that, all future evaluations can focus on investigating the impact of T_{Delay} on cost and QoE at different throughput values of the low-cost path.

5.3 Youtube measurement with static video resolution

After determining the initial COM parameter set, the actual verification of COM focuses on a detailed analysis of the COM-T_{Delay} parameter compared to the CPF principle ($T_{Delay} = 0\,s$) and single usage of the low-cost path ($T_{Delay} = T_{StartDelay} = inf$). With the LTE share – consumption of the high-cost path – and QoE parameters for video resolution r, initial load time $t_{initial}$ and buffering time during playback t_{stall}, the individual gains can be calculated. To cover realistic scenarios, the low-cost path throughput (DSL) is varied in the range 1 Mbps - 100 Mbps. This range corresponds to real DSL deployments and results will show that still higher throughputs will not deliver meaningful insights into the effect of COM. In principle, the LTE access is not a limiting factor as up to 300 Mbps were available during measurements. The service under test was selected to be Youtube as a major VoD provider. Since VoD allows video to be sent with static or adaptive video resolutions, both scenarios are under investigation consuming a 1080p video[3] for 300 s. While in this section the focus is on the static resolution case, in the next section 5.4 the dynamic video resolution case is analyzed.

The measurement time $t_{measurement} = t_{playback} + t_{initial} + t_{stall} = 300\,s$ is long enough to allow a reasonable playback time to show significant results and avoid major outliers. For that, requesting the video on the HTTP layer is included, while establishment of a MPTCP connection (transport layer) is not included. A selection of 1080p resolution is considered fair as this corresponds to typical screen resolutions and available encoding at the VoD providers. According to the MOS approach described in section 2.5, this is the most demanding case. Nevertheless, one can conclude from the following results for 1080p, similar effect of the COM on video transmission with lower video resolution. This is noticeable through higher cost optimization due to similar burst peak rates but less video data (B_{mc}) to be transmitted, ultimately consuming less LTE when COM locks the LTE. Due to the fact that COM is implemented on the MPTCP layer, Congestion Control (CC) using Cubic and BBR applies as recommended in section 2.3.

In the first set of results Figure 5.8, Figure 5.9, and Figure 5.10, the static 1080p Youtube video is evaluated, continuing with a T_{Delay} range $\{0, 0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 5, 7, 10, inf\}$ s. With pure DSL use ($T_{Delay} = inf$), the LTE consumption is understandably not present, and the highest possible t_{stall} and $t_{initial}$ have an impact on QoE. On the opposite, the CPF principle ($T_{Delay} = 0\,s$) with access to the largest throughput has the drawback of the largest LTE usage while QoE results demonstrate the best experience with an interruption free

[3] https://www.youtube.com/watch?v=aqz-KE-bpKQ

playback (t_{stall} = 0 s) and a minimal load time $t_{initial}$ of the video after the request.

In the range 10-100 Mbps, it can be stated that single DSL provides the same QoE as CPF, basically meaning that in the latter case any LTE consumption is spurious demand and therefore unnecessary cost creation. This means that, with $T_{Delay} = \{2,...,4\}$ s, the usage of LTE can be safely reduced to almost zero without impacting QoE. This is a significant gain over the CPF caused LTE usage of 20 %-70 % in the range of 10-100 Mbps. For lower throughputs an increasing T_{Delay} means lower QoE compared to CPF. On the other hand, this is considered acceptable, as it is still much better than DSL only. Looking into the same range of $T_{Delay} = \{2,...,4\}$ s, only $t_{initial}$ causes some longer loading time of the video, while LTE consumption goes down to in maximum one eight (6 Mbps), a half (3 Mbps) or three quarter (2 Mbps). If for example the initial loading time is considered less relevant for the overall QoE COM can be used without concerns also at this lower DSL throughputs taking the advantage of cost reduction with minimal QoE impact compared to CPF.

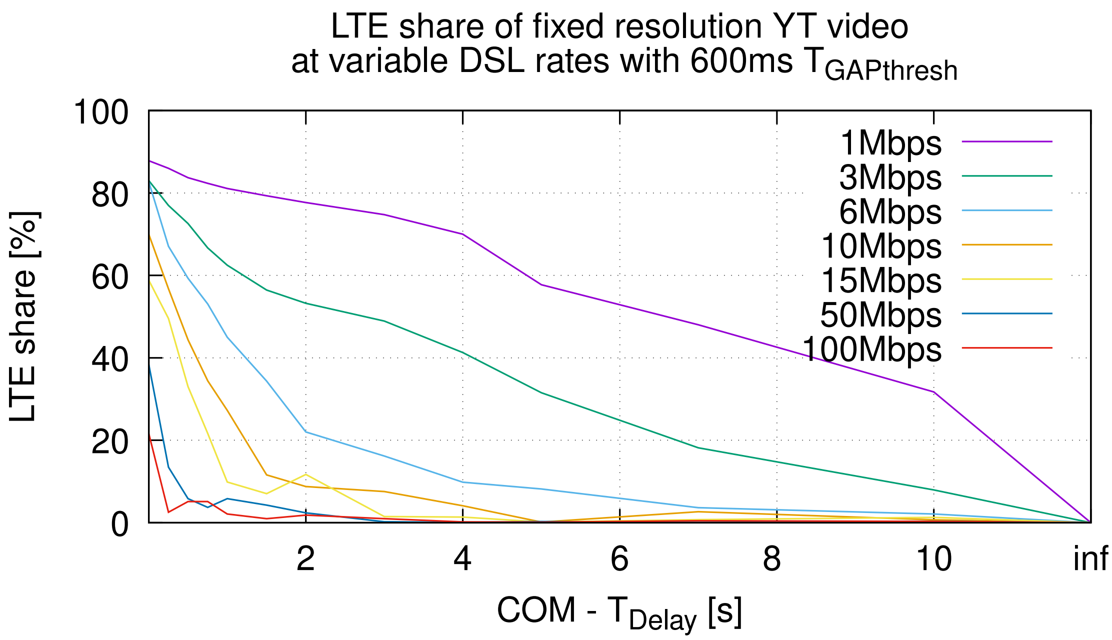
These observations are mainly for Cubic but BBR results show the same trend, even if the LTE consumption has a minimal flattened slope, which leads to a QoE only impacted at higher T_{Delay}.

When this is further explored in terms of the gains defined in section 4.2, the visualised results provide enough information to evaluate them without calculating them individually.

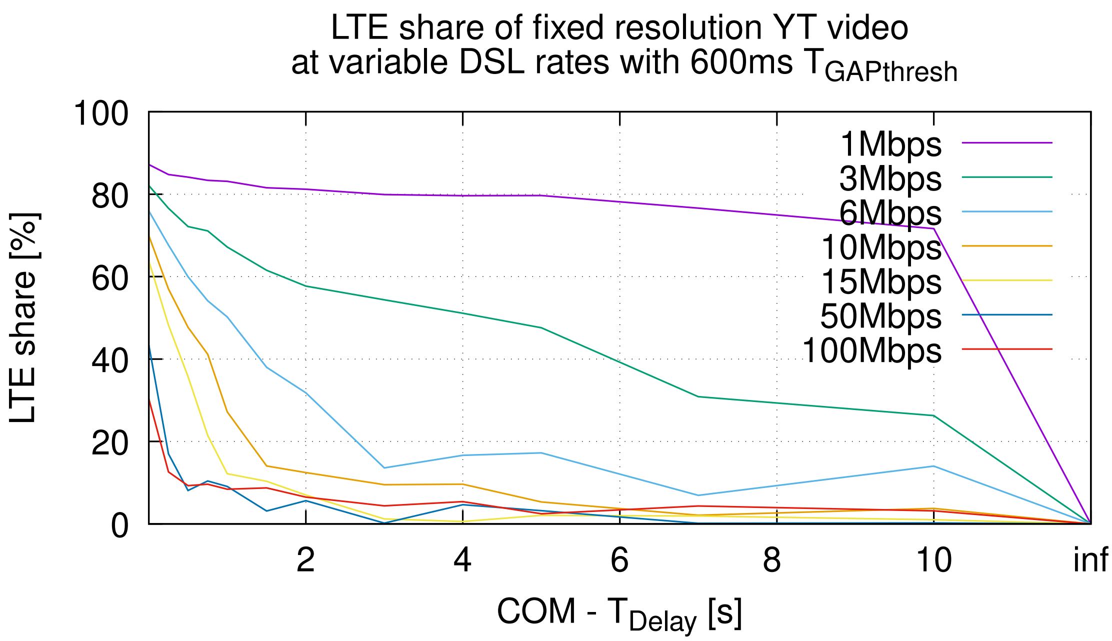
Considering again the QoE irrelevant range of 10-100 Mbps, the cost factor $\Delta\theta$ ranges from 0 % – no cost – and the negative value of -70 % – highest cost. This is justified in the definition of $\Delta\theta$ in Equation 4.11, which on the one hand cannot be greater than zero and on the other hand is determined by the negative maximum of $|\theta_{SCP} - \theta_{MP}|$ or $|min(\theta_{MP,A1}, \theta_{MP,A2}) - max(\theta_{MP,A1}, \theta_{MP,A2})|$. With the lowest cost of zero in the SCP scenario and the highest cost determined by the LTE share of 70 % in the 10 Mbps CPF, the above range between -70 % and 0 % is obtained.

Within the interesting range of $T_{Delay} = \{2,...,4\}$ s also identified above, this cost factor is pushed to zero or close to zero, the desired ideal. When the lower DSL throughputs below 10 Mbps are included, the cost factor reaches the -88 % range with the potential to decrease with increasing T_{Delay} but not eliminate the cost.

Something similar can be found for the gain γ. Since the video resolution is static, a consideration can only be made for $\gamma_{initial}$ and γ_{stall}. For the range of 10-100 Mbps and $T_{Delay} = \{2,...,4\}$ s, both values are either $\gamma = 1$ – unchanged QoE – compared to CPF or $\gamma > 1$ – better QoE – compared to SCP. At lower DSL throughputs the gains γ might fall

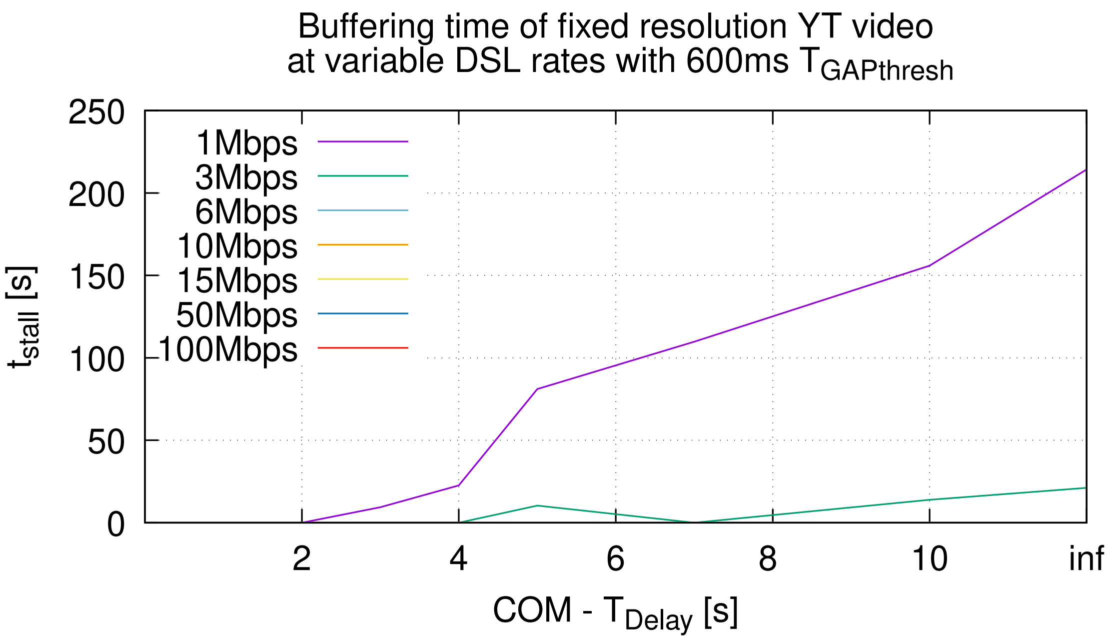


(a) Cubic

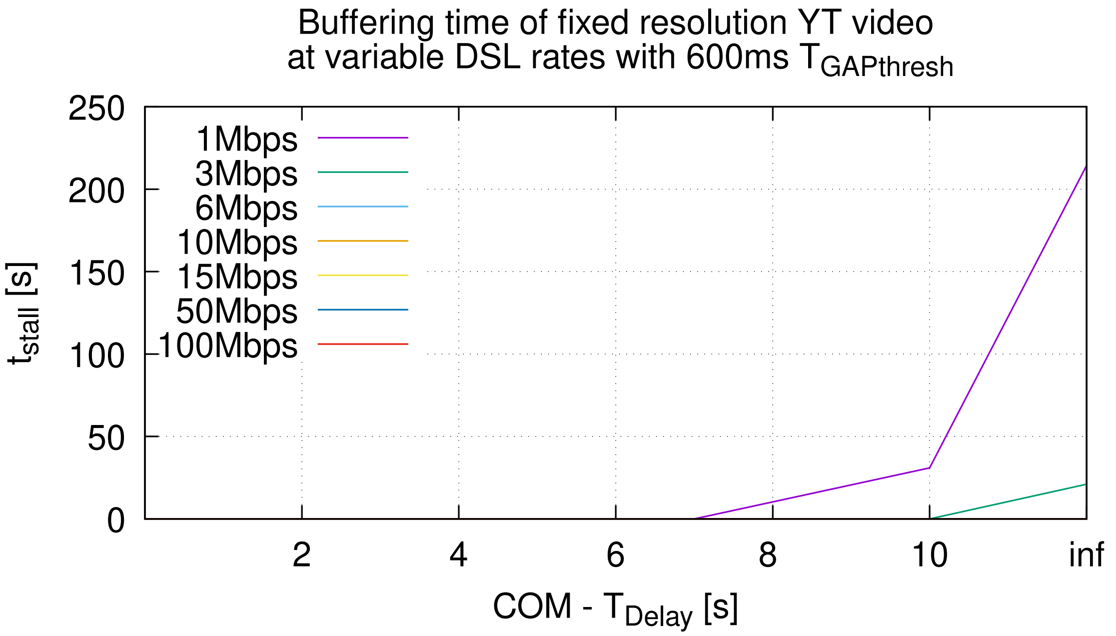


(b) BBR

Fig. 5.8 LTE share of static 1080p YT video at variable DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$

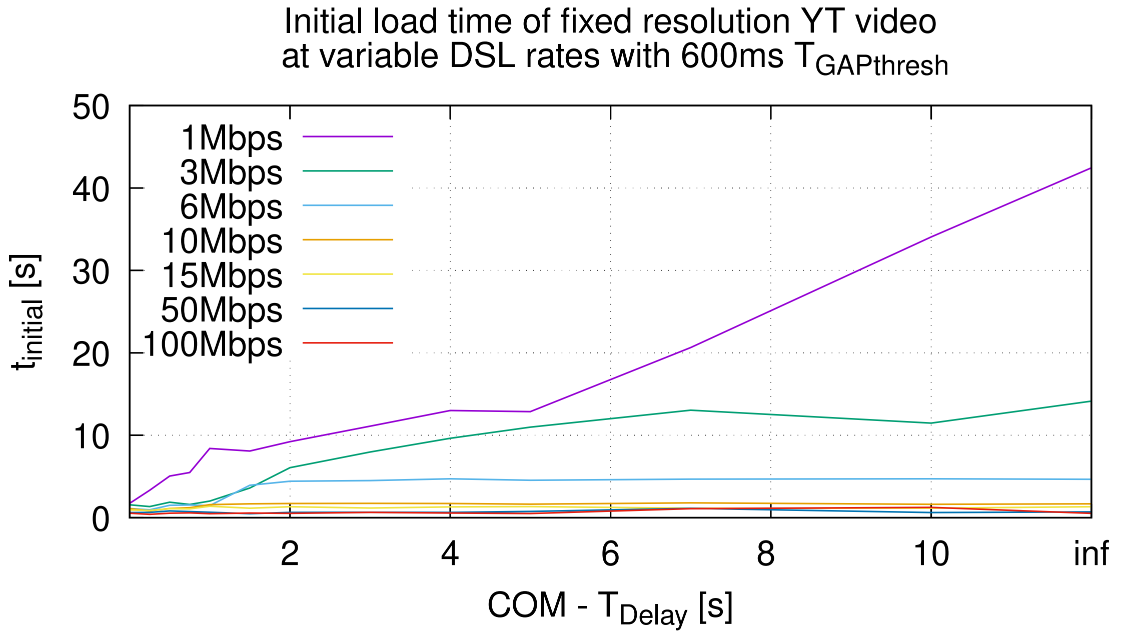


(a) Cubic

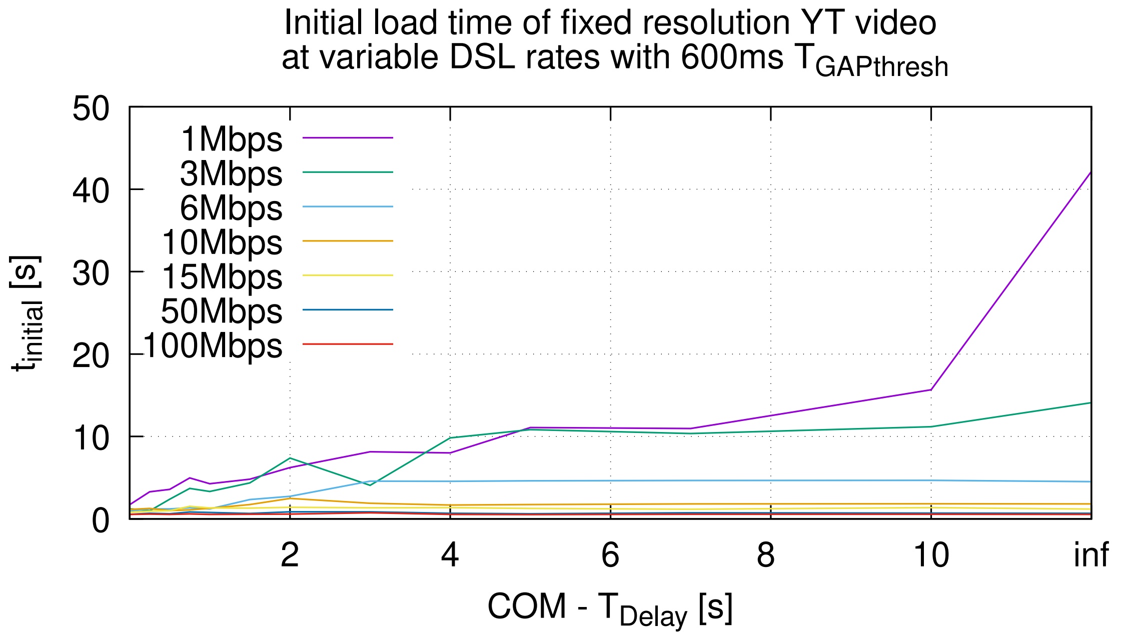


(b) BBR

Fig. 5.9 Buffering time of static 1080p YT video at variable DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$



(a) Cubic



(b) BBR

Fig. 5.10 Initial load time of static 1080p YT video at variable DSL rate, $t_{GAPthresh} = 600ms$, $t_{notraffic} = 50ms$ and $V_{max} = 100pkts$

below 1 depending on the used CC if compared to CPF otherwise the gain is 1 or higher if compared to higher T_{Delay} or DSL only.

5.4 Youtube measurement with adaptive video resolution

In the case of video streams which adapt the resolution according to the available throughput, it is particularly exciting to evaluate the interaction with COM. The setup remained unchanged with regards to the measurements of the static resolution video case, even the same video sequence has been used. The only difference was the activation of the adaptive video resolution which is typically the default setting. In this more complex scenario, the representation of QoE results moves away from line graphs to bar diagrams as this allows the different shares of the video resolutions (144p, 240p, 360p, 480p, 720p, and 1080p) to be captured and the loading times $t_{initial}$ and t_{stall} during playback. A bar represents $t_{measurement}$ with the different shares of $t_{initial}$, t_{share}, and $t_{playback}(r)$ with video resolution $r = \{144p, 240p, 360p, 480p, 720p, 1080p\}$. The video resolution of 144p is a resolution that is usually not taken into account in the calculation of the MOS derived QoE from section 4.2. However, since Youtube uses this resolution as a lower limit for video playout, it is included though and modifies Equation 4.6 to add to the vector a weighted time $t_{144\,\mathrm{px}} \cdot 10^{-5}$.

The LTE consumption shown in Figure 5.11 is mostly similar to the static video resolution results in Figure 5.8, thus indirectly confirming these. Without T_{Delay} configured – CPF behavior – the LTE share is identical, which is confirmed by the streamed video resolution of 1080p shown for Figure 5.12 (1 Mbps), Figure 5.13 (3 Mbps), and Figure 5.14 (6 Mbps). As CPF does not limit the available throughput, 1080p video streaming applies, as confirmed by the almost fully blue bars at $T_{Delay} = 0\,\mathrm{s}$. For all further T_{Delay} which show larger blue bars – 1080p streaming – the composition of the lines follow the results of the static video resolution playback. This is especially true for the DSL throughput $\geq 6\,\mathrm{Mbps}$, which is shown with all bars blue at any T_{Delay} in Figure 5.15, Figure 5.16, Figure 5.17 and Figure 5.18. Differences are mainly visible for the 1 Mbps and 3 Mbps results. In both cases, video resolution falls below 1080p which leads to a greater drop in the LTE share compared to static video resolution. But even if the video resolution adapts to the lower values at certain T_{Delay}, 3 Mbps results at least provide resolution of 720p for $T_{Delay} > 3\,\mathrm{s}$. In the case of 1 Mbps this has higher dependency on Congestion Control, but a drop to 720p happens for Cubic for $3\,\mathrm{s} < T_{Delay} \leq 4\,\mathrm{s}$ followed by drops to 480p, while BBR is even more vulnerable to lower T_{Delay} showing shares of 480p earlier. Most likely in this limited range COM interferes with BBR's mechanism to find the optimal operation point. Another remarkable development compared to the static video resolution case is that t_{stall} and $t_{initial}$ are quite low. The adaptive resolution mechanism reduces stalling and initial loads to a minimum - if necessary, this is compensated by a lower resolution. COM does not seem to change this in any way.

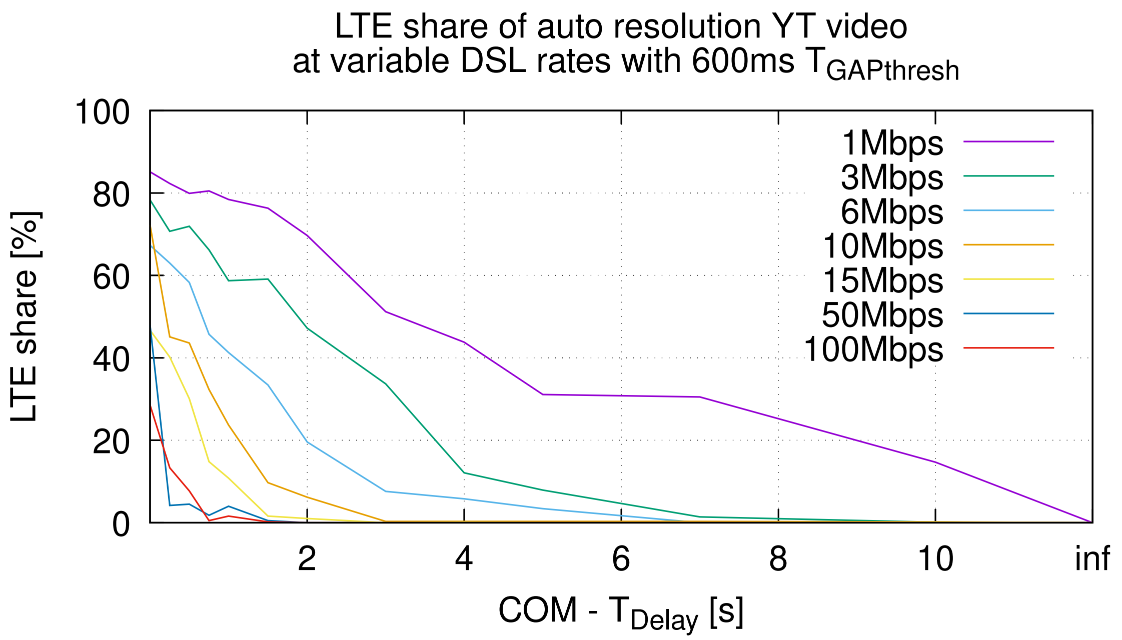
Following the cost factor analysis in section 5.3, the factor does not deviate significantly. As described above the LTE consumption shown in Figure 5.11 is in large areas the same compared to Figure 5.8, especially in the identified QoE irrelevant area of the static video resolution trial in the range of 10-100 Mbps. For lower DSL throughputs, a statement made in section 5.3 can also be confirmed that video resolutions below 1080p increase the cost optimisation potential. The gradient in the dynamic video resolution measurements is steeper, which is due to the ABR algorithm choosing lower video resolutions with lower B_{mc} demand.

The picture is reversed when the gain is considered in Figure 5.12 - Figure 5.18. While in the measurements with static video resolution $\gamma_{initial}$ and γ_{stall} were in the foreground and $\gamma_{resolution}$ irrelevant, this is now changed. Due to the objective of ABR, $t_{initial}$ is consistently kept at a very minimum level and t_{stall} is not existent due to the preferred usage of lower video resolution before a stall events can happen. Both gains $\gamma_{initial}$ and γ_{stall} are therefore 1 across all DSL throughput measurements with SCP, CPF and COM. This lets the focus on the experienced video resolution. Wherever the results show a fully blue bar, the $QoE_{resolution}$ (Equation 4.6) is equal to 1. With a generous view it can be stated that as of 6 Mbps but latest with 10 Mbps DSL throughput $\gamma_{resolution} = 1$ independent of SCP, CPF or COM. For lower throughputs $\gamma_{resolution}$ is > 1 if compared with SCP and < 1 if compared with CPF and lead to a predominant use of 720p in the range of $T_{Delay} = \{2, ..., 4\}$ s which was identified interesting in the static video resolution measurements.

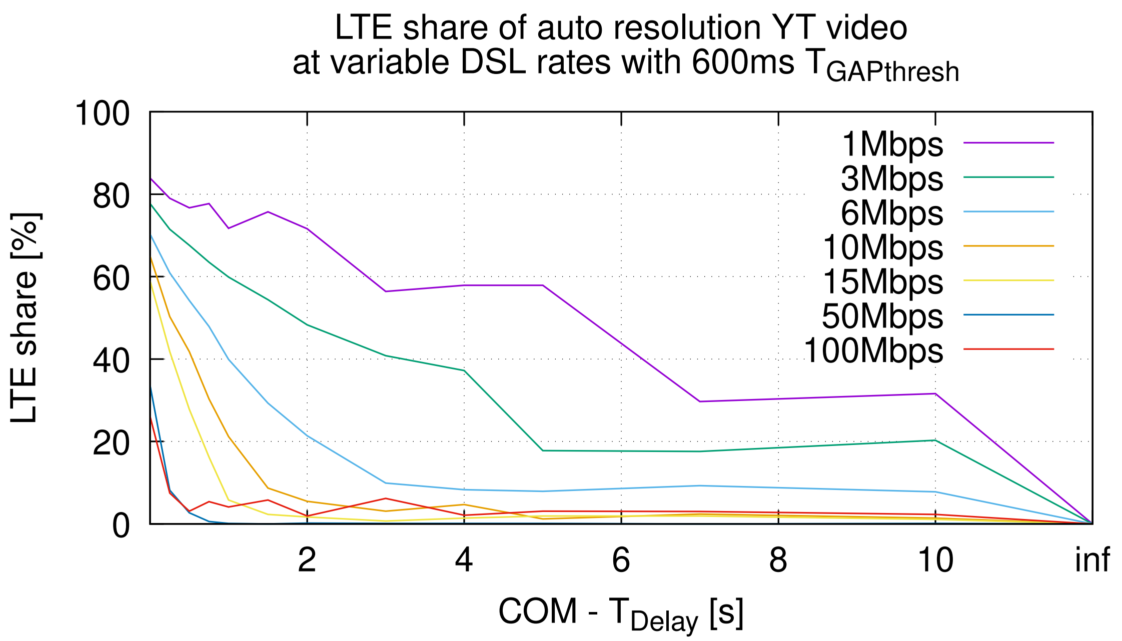
To emphasise the understanding of the results from a cost perspective:

If DSL-only is shown blue, CPF cannot make it better.

This basically means when the best QoE can already be provided over the single low-cost path – the DSL path –, CPF has no advantage. Considering, however, the high CPF induced LTE share across the DSL throughput range of 1-100Mbps, significant costs are raised without any benefit. COM scheduler addresses this by lowering or removing the LTE share when Hybrid Access (HA) is used to provide connectivity, while not adding any QoE advantage over the single DSL path use.

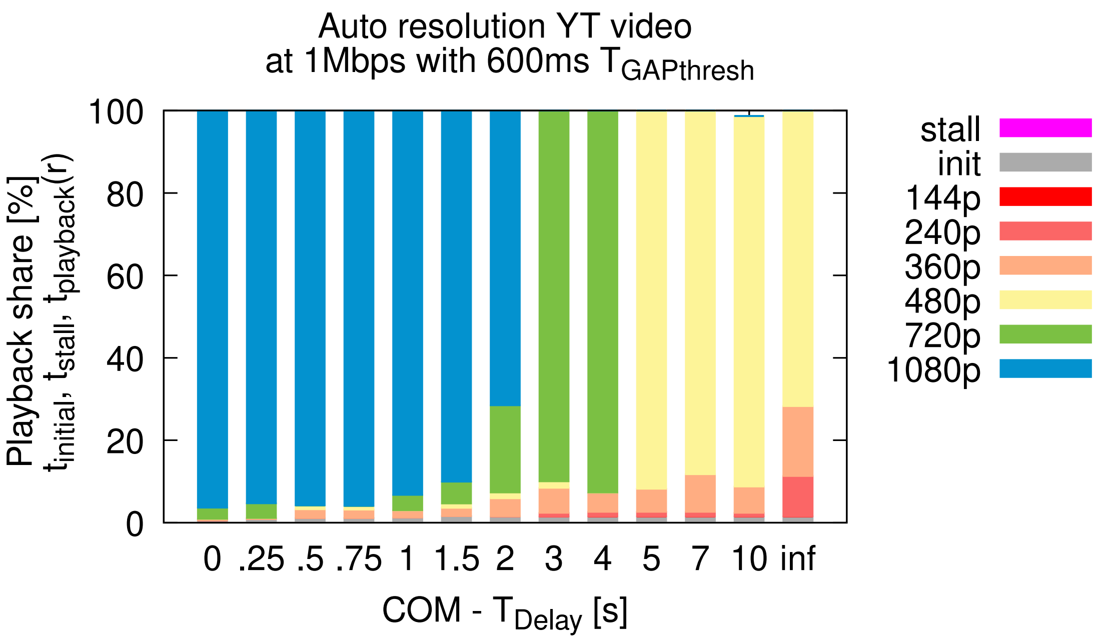


(a) Cubic

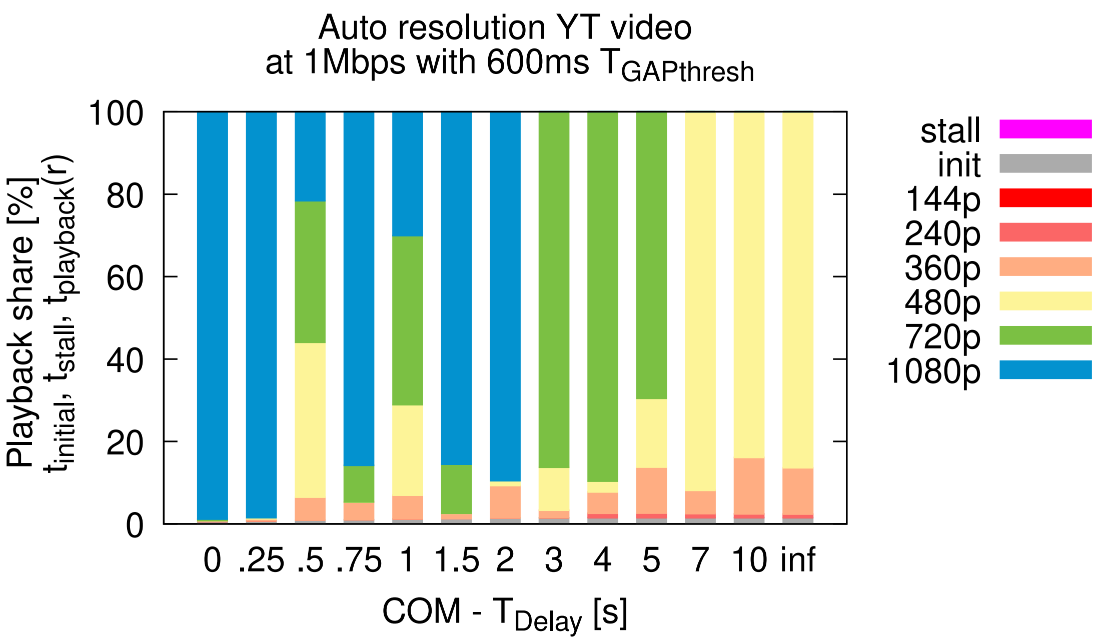


(b) BBR

Fig. 5.11 LTE share of adaptive resolution YT video at variable DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$

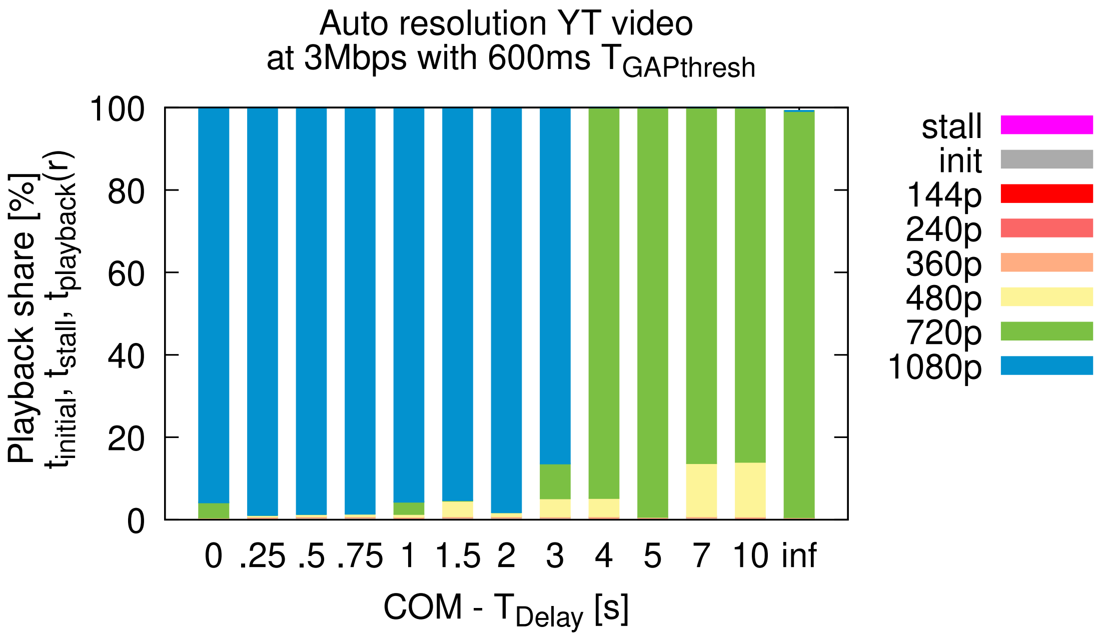


(a) Cubic

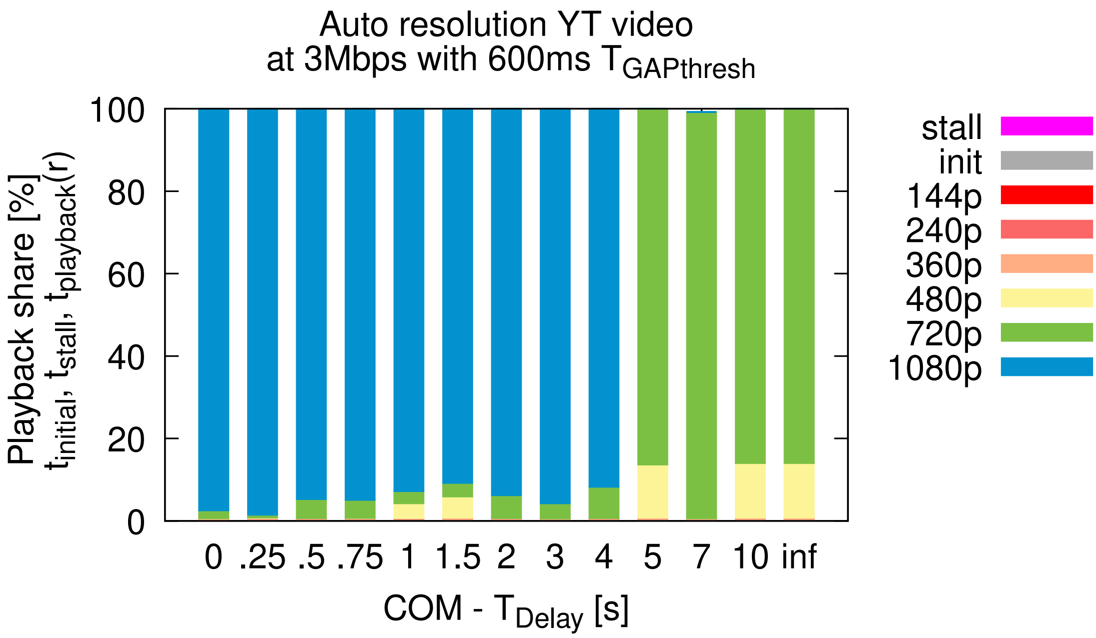


(b) BBR

Fig. 5.12 QoE parameters of adaptive resolution YT video at 1 Mbps DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$

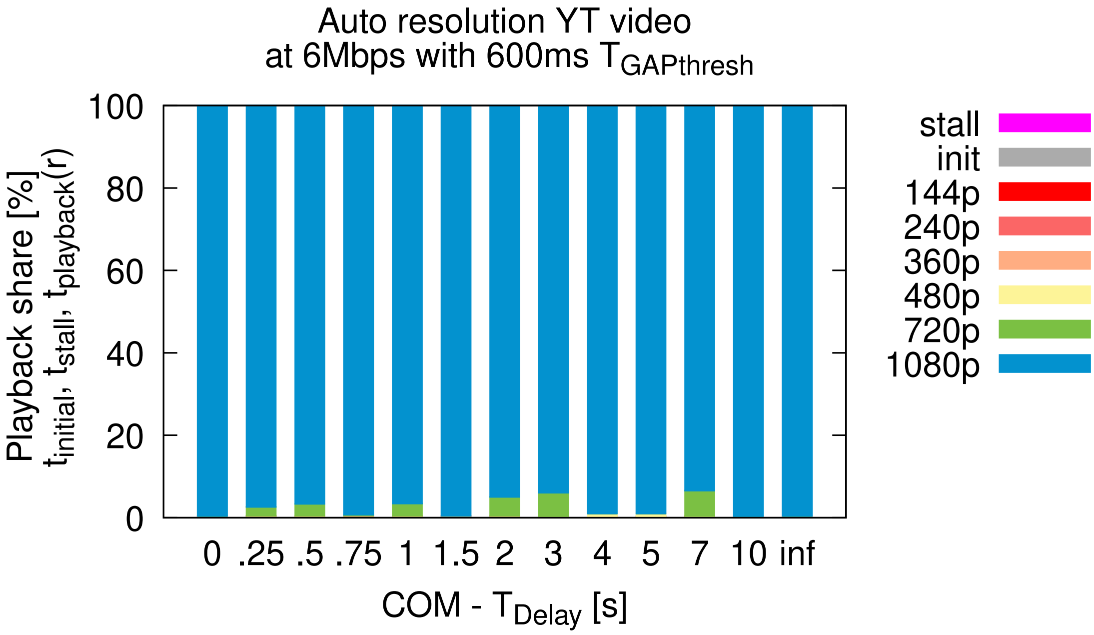


(a) Cubic

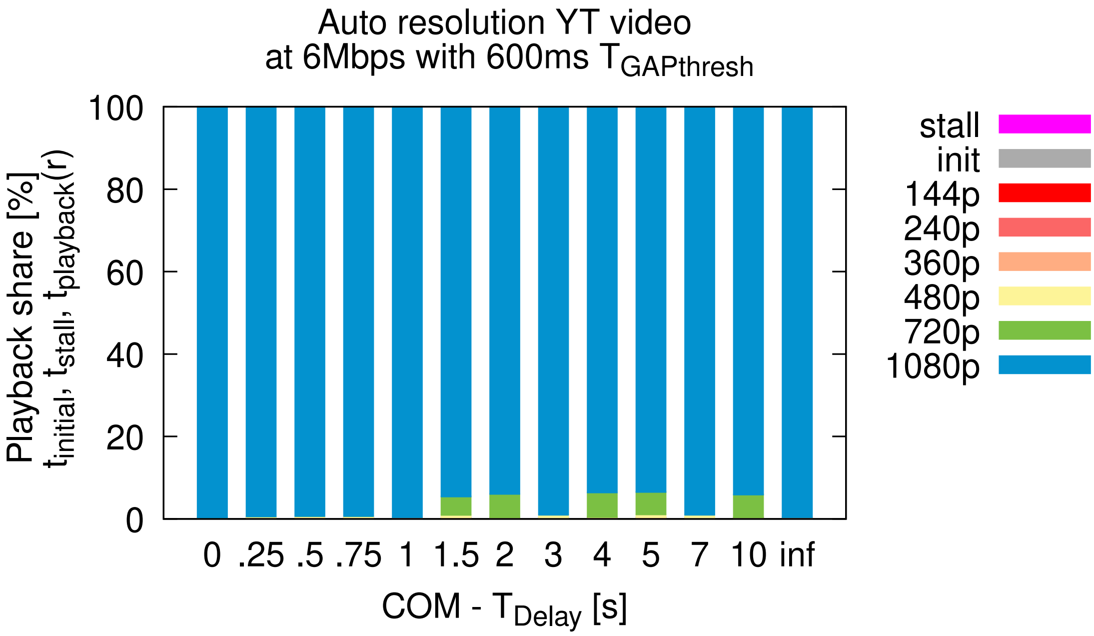


(b) BBR

Fig. 5.13 QoE parameters of adaptive resolution YT video at 3 Mbps DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$

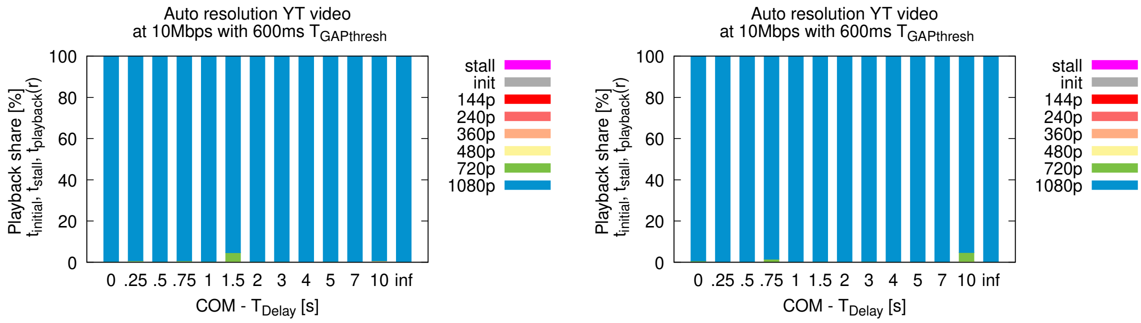


(a) Cubic



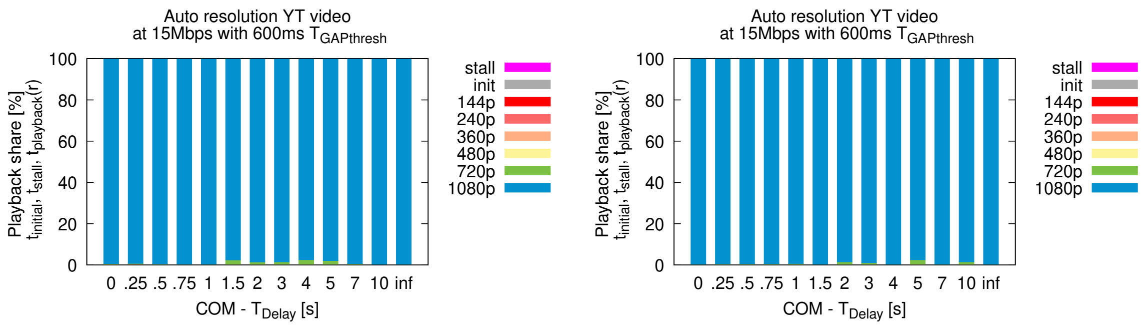
(b) BBR

Fig. 5.14 QoE parameters of adaptive resolution YT video at 6 Mbps DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$



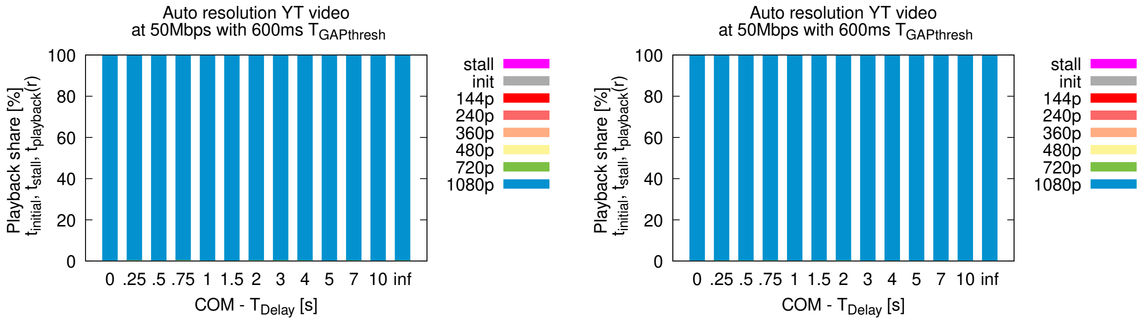
(a) Cubic (b) BBR

Fig. 5.15 QoE parameters of adaptive resolution YT video at 10 Mbps DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$



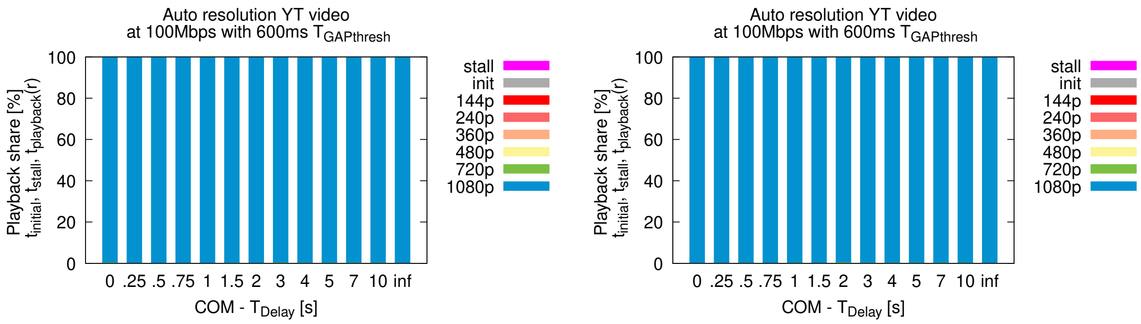
(a) Cubic (b) BBR

Fig. 5.16 QoE parameters of adaptive resolution YT video at 15 Mbps DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$



(a) Cubic (b) BBR

Fig. 5.17 QoE parameters of adaptive resolution YT video at 50 Mbps DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$



(a) Cubic (b) BBR

Fig. 5.18 QoE parameters of adaptive resolution YT video at 100 Mbps DSL rate, $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$

The main conclusion from the QoE perspective can be defined as:

If DSL-only is not blue, COM provides always better QoE up to the CPF level.

In the tested range of COM parameters, COM (and eventually CPF) always provides better experience compared to single-path DSL. Depending on the direction from which one wants to optimise, either the CPF QoE, or the single-path DSL QoE can be defined as the benchmark. If selecting the latter, a T_{Delay} of 4 s seems to be efficient, as it provides across all DSL throughputs at least 720p video resolution (except some 480p at BBR 1 Mbps), and when DSL provides more than 3 Mbps, the resolution goes up to 1080p. In contrast, the cost is cut by a half for 1 Mbps DSL when Cubic CC is used and to about one third when BBR is used. For DSL at 3 Mbps, the costs for Cubic are close to zero, while the same trend is observed for BBR, albeit with a slight shift from 6 Mbps onwards. Also a T_{Delay} of 4 s is appropriate if contrary, the CPF QoE with almost always 1080p video resolution is the benchmark, at least for the DSL throughput greater than 3 Mbps. For the other throughput values, smaller T_{Delay} values are required. For example, $T_{Delay} = 2$ s for throughput 3-6 Mbps with a cost cut by almost a half at 3 Mbps and for the lower throughput towards 1 Mbps $T_{Delay} < 1$ s, with minimal cost reduction.

5.5 Impact on website requests

The potential of COM with respect to the main objective of this work, to handle VoD traffic more efficiently than CPF, has been demonstrated in section 5.3 and section 5.4. Since COM aims to replace CPF, it is worthwhile to understand the implications for other scenarios that would allow general use beyond VoD. With COM's MPTCP implementation, one is still tied to TCP services, which give rise to another important usage scenario: Website calls. These calls are usually of short duration with the aim of rendering a website as quickly as possible in the calling application (browser). What makes it interesting is the relationship to VoD. Due to the usage of TCP, the transmission strives to achieve maximum possible transmission rate which will also create a burst as with VoD. The difference is that the burst of website calling is only once which let COM not detect any gap between consecutive burst. However, the $T_{StartDelay}$ parameter is designed for this scenario to change the cost ratio already at the beginning of a connection without requiring gap detection. For the testing with VoD traffic $T_{StartDelay}$ was set already to $T_{StartDelay} = T_{Delay}$. This effect has now been investigated with three different websites `ibm.com`, `tagesschau.de` and `telekom.de` in July 2018. The selection criteria was arbitrary but driven by media heavy content to pronounce QoE impacts. An overview in Table 5.1 of the relevant characteristics of the web pages in the test scenario shows a transmission volume in the range between 2 - 4 Mbps, depending on the dynamic content of the web page. The number of connections also varies, which ultimately affects the number of MPTCP connections involved. Even if HTTP 2.0 was used, which routes the resource request and transfer over a single HTTP connection, this only affects the resources that belong to the same domain. For example, if fonts or javascript files are hosted in different locations, this will result in separate HTTP requests. Nevertheless, it was ensured by measurement that the main transmission ran over one connection.

Similar to the tests with VoD also here the Browser plugin `uBlock Plus Adblocker` was used to avoid too much interference of unpredictable content delivery which impacts the transmitted volume and in particular the time of loading the website.

Website	Size min / avg / max	Connections min / avg / max
`ibm.com`	2.19 / 2.85 / 3.91 MB	3 / 14 / 18
`tagesschau.de`	2.29 / 3.16 / 4.38 MB	11 / 13 / 19
`telekom.de`	2.03 / 2.48 / 3.47 MB	10 / 15 / 25

Table 5.1 Relevant website characteristics

Since the focus is on calling websites, *Cubic* was chosen, which is the standard Congestion Control (CC) of the most widely used operating systems (see subsection 2.3.1).

Chromium browser was used in version 64.0.3282.119 (Developer Build) built on Debian 9.3 as described in section 4.3 to generate the requests and collect the necessary data.

COM uses the parameterization set in section 5.2, while the only relevant part, as explained above, is the setting $T_{StartDelay} = T_{Delay}$, since the gap detection and subsequent delay of the expensive path never takes place.

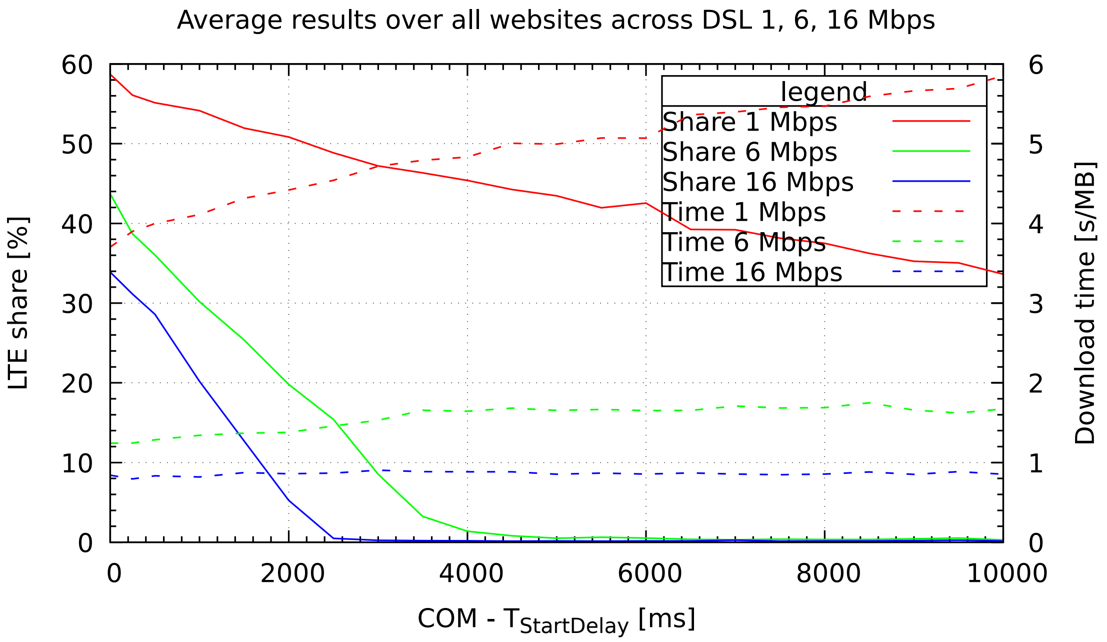
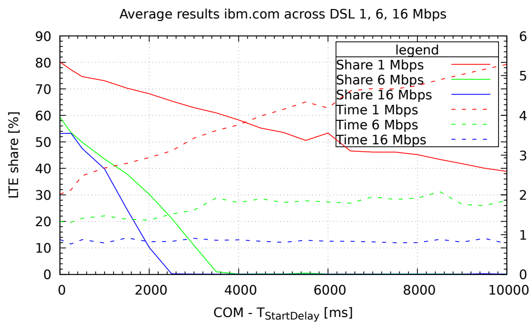


Fig. 5.19 Average LTE share and normalised download time over three different website requests with variable $T_{StartDelay}$ across 1, 6 and 16 Mbps DSL

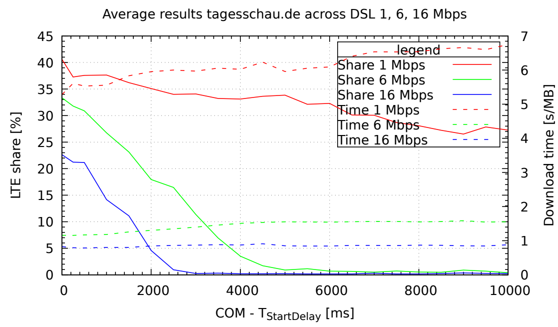
The measurement and analysis approach in terms of cost remains unchanged compared to the measurement of VoD, calculating the amount of traffic sent over the expensive path (LTE) compared to the low-cost path (DSL) – LTE share – plotted over the $T_{StartDelay}$ parameter under the constraint of different DSL throughput.

These boundary conditions also apply to the presentation of QoE, but the meaning has changed compared to VoD and the download time is used instead. It is known that instead of the download time, there is also a time, usually earlier, which designates the time at which a first contentful paint of the website takes place and could thus be an alternative for determining the QoE. However, the trend in costs and QoE of interest here is not affected by this.

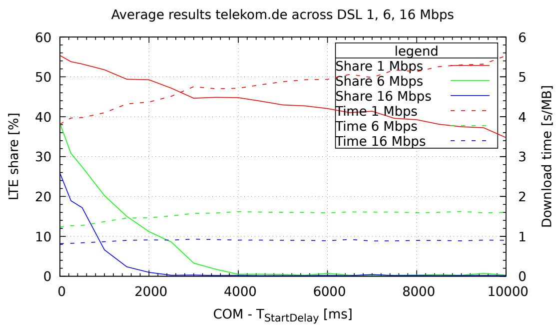
To avoid fluctuating loading times due to the variable amount of data (see Table 5.1) when calling up the website, the loading time is normalised by dividing the amount transferred.



(a) ibm.com



(b) tagesschau.de



(c) telekom.de

Fig. 5.20 Per website LTE share and normalised download time over variable $T_{StartDelay}$ across 1, 6 and 16 Mbps DSL

The results are not surprising and in large areas share similarities with the VoD results. As shown in Figure 5.19 for an average of all the three websites across DSL rates of 1, 6 and 16 Mbps and in Figure 5.20 broken down to the individual websites, a huge LTE share is caused when CPF ($T_{StartDelay} = 0$ sec) is used, which can be reduced or even eliminated with increasing $T_{StartDelay}$. Except for the 1 Mbps case, the LTE share elimination range also moves between 2-4 s $T_{StartDelay}$. An effect that does not play a role in the VoD measurement, or at least is not visible due to the scaling, becomes visible here when looking at QoE and already shows differences in the normalised download time across the DSL throughputs in the CPF scenario. This is related to the design of MPTCP which first establish an initial subflow before (see `MP_CAPABLE` and `MP_JOIN` procedure in [10, Sec. 2.1 and 2.2]) after earliest 2RTT a subsequent flow can be used for data transmission. Due to the noticable short measurement period for calling a website compared to the 300 s for consuming a video and the scale of seconds, this almost lead to a double of the normalized download time when 6 Mbps is considered compared to 16 Mbps where the download time runs into a saturation and is not impacted anymore by LTE availability. That is also the reason why measurements at higher DSL rates are not shown since the QoE does not change beyond 16 Mbps and elimination of LTE share starts at earlier $T_{StartDelay}$.

Overall, these QoE results might look in the beginning not very promising and it cannot be whitewashed the fact that the download time is doubled from 16 Mbps compared to 6 Mbps and again doubled compared to 1 Mbps as of $T_{StartDelay} = 4$ s, but several aspects in the following list put this into perspective again to a certain extent.

1. Due to MPTCP procedure a download time offset is introduced which becomes higher as lower the DSL throughput is. For a gain calculation this needs to be taken into account.
2. The normalized download time is here considered for QoE and not the absolute download time. However, the absolute time is relevant for the user experience. For low-volume websites, the absolute download time remains low even when doubled.
3. As raised above also a first contentful paint time exists which usually is lower than the download time. After this time first content is visible and user can start to interact with the website. This further reduces the impact of increased download time.
4. Compared to single path QoE which is represented by $T_{StartDelay} = 10$ s, most often an improvement is possible depending on the selected $T_{StartDelay}$ but at least no worse behaviour emerges.
5. Only lower DSL rates are affected at all by QoE variances.

Similar to the findings in section 5.3, the cost factor $\Delta\theta$ moves from $-80\,\%$ occurring at 1 Mbps CPF down to $0\,\%$ under the use of COM. For DSL rates as of 6 Mbps this minimal cost is gained when $T_{StartDelay} = 4$ s is selected and leads to a cost drop of a quarter at 1 Mbps.

For the gain of QoE no definition is provided in section 4.2 which has the focus on VoD but at least the same interpretation of γ can be reused. In principle it can be stated that $\gamma = 1$ for DSL rates at and beyond 16 Mbps which means any evaluated COM setting did not impact the QoE. If looked at the 6 Mbps line in Figure 5.19 and take the offset into account the gain from a CPF perspective to SCP is $\frac{1.2\,\mathrm{s}}{1.6\,\mathrm{s}} = 0.75$ which is below 1 and therefore worse compared to CPF. This is also the case at the 4 s $T_{StartDelay}$ demarcation line which seems overall to be quite promising. A similar calculation applied for the 1 Mbps line results in $\frac{3.6\,\mathrm{s}}{6\,\mathrm{s}} = 0.6$ which is, however, at the demarcation line $\frac{3.6\,\mathrm{s}}{4.8\,\mathrm{s}} = 0.75$. No need to mention that the gain is higher if the same calculation is carried out from the point of view of SCP.

It is obvious that these absolute numbers are based on a few measurements of which no general validity can be claimed. On the other hand also these results are very much inline with the expectations about COM's mode of action formulated in section 3.7 and the previous measurements and results in this chapter.

However, in order to get a sense of the real user experience, it is essential to test it in front of users.

5.6 Customer trial

Many results presented in this chapter show a positive or most often significant effect of COM on cost reduction and a QoE impact which is either unchanged compared to CPF or in an assumed acceptable region. The movement in this two dimensional space reflected in Figure 4.1 is finally controlled by COM's $T_{Delay} = T_{StartDelay}$ parameter and dependent on the throughput capabilities of the cheap path. These results are now looking for an ultimate confirmation that can only be achieved through real-life use if they are verified by users.

As part of a larger year-long multi-connectivity trial program for a Tier 1 operator in Europe, the opportunity arose in 2020 to get exactly this confirmation. Corresponding to the ATSSS testbed architecture shown in Figure 4.9 COM, along with CPF and the default *minRTT* scheduler was implemented in Google Pixel 2 devices handed over to customers of this operator.

The idea behind this activity was to get unbiased feedback from everyday users about their perceptions of daily usage, while measuring the distribution of traffic when three scheduling mechanisms *minRTT*, *CPF*, *COM* are in place for one month each. In order to ensure a smooth process, several intelligent mechanisms have been introduced that allow for high-quality results:

1. Users received a Google Pixel 2 mobile phone as their first/main device with an AOSP-based OS with all relevant services known from standard devices, such as the App Store. Enquiries in advance ensured that the users had mobile phones from which a migration to the new device was possible while retaining the user experience. In a 1 to 1 support, the migration was carried out with the user at the handover of the new device.

2. 1 month at the beginning of the trial without any changes to the network stack, so that the user can get used to the new device without the results being affected by the acclimatisation phase.

3. Cellular data plan without volume restrictions – Full flat

4. Remote control of multipath settings to switch phases without user perception.

5. Best multipath experience which guarantees connectivity whenever at least one access is available (Figure 5.21). This is more detailed below.

6. Suppression of QUIC traffic to obtain maximum possible coverage of services using MPTCP. In 2020, this was possible by only blocking UDP port 443 and increasing a measured previous 80 % share of TCP back to 99 % without any degradation of service due to service integrated TCP fallback mechanisms.

7. In the 3-month trial period of MPTCP activated, user could not deactivate the Wi-Fi and cellular interface. This was achieved by manipulating the user interface to prevent accidental switching off of an interface, which would also have meant switching off multi-connectivity. The assumption behind this implementation was that users don't actually see a reason to switch the network interfaces manually unless they have a connectivity pain point that makes them think they have to do something. If one prevent this through a smart multipath concept, there is no need. With the concept shown in Figure 5.21 this was achieved and users never noticed during the trial phase that they could not switch the network interfaces.

8. Along with the previous point the devices were forced to login to the user's home Wi-Fi, all stored Wi-Fi of the old device of the user and to the Wi-Fi hotspot infrastructure of the Tier 1 operator. This measure helped to increase the time of multipath usage.

9. Regular survey of the study participants to find out about the user experience.

As mentioned in the list above, the concept presented in Figure 5.21 ensures multipath is integrated and used in a way that can be expected by a user. In short: All traffic is handled and connectivity is always usable if at least one access is available. What sounds like a given requires a lot of sophisticated adjustments when MPTCP is used as the basis for multi-connectivity. As the concept has been tested in different countries and user groups and with different objectives, it contains elements that have not all been activated within the scope of this work. In the trial setup of Figure 4.9, this concept is implemented in the smartphone and the MPTCP Proxy to handle both directions of traffic, the down- and the uplink traffic.

The heart of this concept under investigation is the MPTCP scheduler which can be configured in the trial to be *minRTT*, *CPF* or *COM*. In case *CPF* or *COM* are activated, the path cost or prioritization was configured to Wi-Fi being preferred because it is considered low-cost and the cellular access to be non-preferred because of considered high-cost. Before traffic reaches the scheduler, a multi-level separator concept ensures that only traffic that is part of an existing (MP)TCP connection passes the scheduler, while any other traffic is forwarded to another separator that filters for QUIC traffic (UDP/443) to block this traffic and finally forwards the remaining traffic to the OS forwarding layer to be transmitted via the default gateway (GW). This includes for example services which rely on UDP or those traffic which is used to establish a (MP)TCP connection. More or less the usage of the default GW corresponds to the status quo without multipath, however three crucial details help to achieve a much better multipath experience:

1. Prioritize any traffic which is relevant to establish a (MP)TCP connection, to confirm data reception in TCP and monitor a connection using TCP keepalive. The same prioritization also occurs to the meta traffic used by the UDP tunnel protocol which is established per access between smartphone and Proxy (see subsection 4.1.3).

   This is ensured by an access and traffic classifier that groups traffic into payload traffic assigned to Wi-Fi by the scheduler, payload traffic assigned to cellular and traffic for meta and signalling purposes. The latter is finally prioritised for demultiplexing for forwarding via the accesses and helps in the following situations:

   - Establishing an initial or subsequent MPTCP flow when an access path is congested. Without this, it was often observed that no MPTCP connection could be established or the subsequent path failed, rendering the idea of multipath useless, especially if only the path pointed to by the default gateway is loaded and the other path is not.

- Early detection of a dead MPTCP-subflow using the TCP-keepalive function to trigger a re-establishment of a subflow in order to extend the time in which multipaths can be used.
- Reliable acknowledgement of packets, especially when the access direction is congested, used to send TCP ACK packets. This helps to make better use of bandwidth by stabilising the congestion window.
- Early detection of an access interruption or availability in order to optimize the handling of traffic which is not belonging to an established MPTCP connection. This pays very much into the configuration of the default Gateway explained in the next item of the list. Traffic handled by the MPTCP scheduler uses TCP's own mechanisms to detect complications with an access and is not affected by the default Gateway setting.

2. Most agile default Gateway configuration which is not exclusively dependent on the detection of a physical layer interruption and typically helps to earlier detect an unresponsive default Gateway path and switch the default Gateway to the other access. The change of the default Gateway on the one side, e.g. smartphone, is also communicated to the other side, e.g. Proxy, to apply the same setting here. This allowed for seamless continuation in both directions of non-TCP traffic communication and increased reliability when establishing a MPTCP connection.
3. Cellular network standard based activation of MPTCP splitting (aggregation) or switching (handover).

   This is based on the observation that splitting MPTCP over Wi-Fi and a 2G or 3G mobile access results in poor overall performance, as the path characteristics are very different and cannot be compensated by reasonable TCP tuning. However, to keep at least the benefit of session continuation when Wi-Fi interrupts or returns, MPTCP is put into the so called backup-mode[4].

The bandwidth shaper shown at different locations in the concept are irrelevant for this work and served other objectives within the mentioned 1-year trial of the Tier 1 operator.

The overall trial period included December 2019 as the acclimatisation period, from mid-January 2020 with activated *COM*, then *CPF* from mid-February and finally *minRTT* from mid-March.

[4]https://multipath-tcp.org/pmwiki.php/Users/Tools

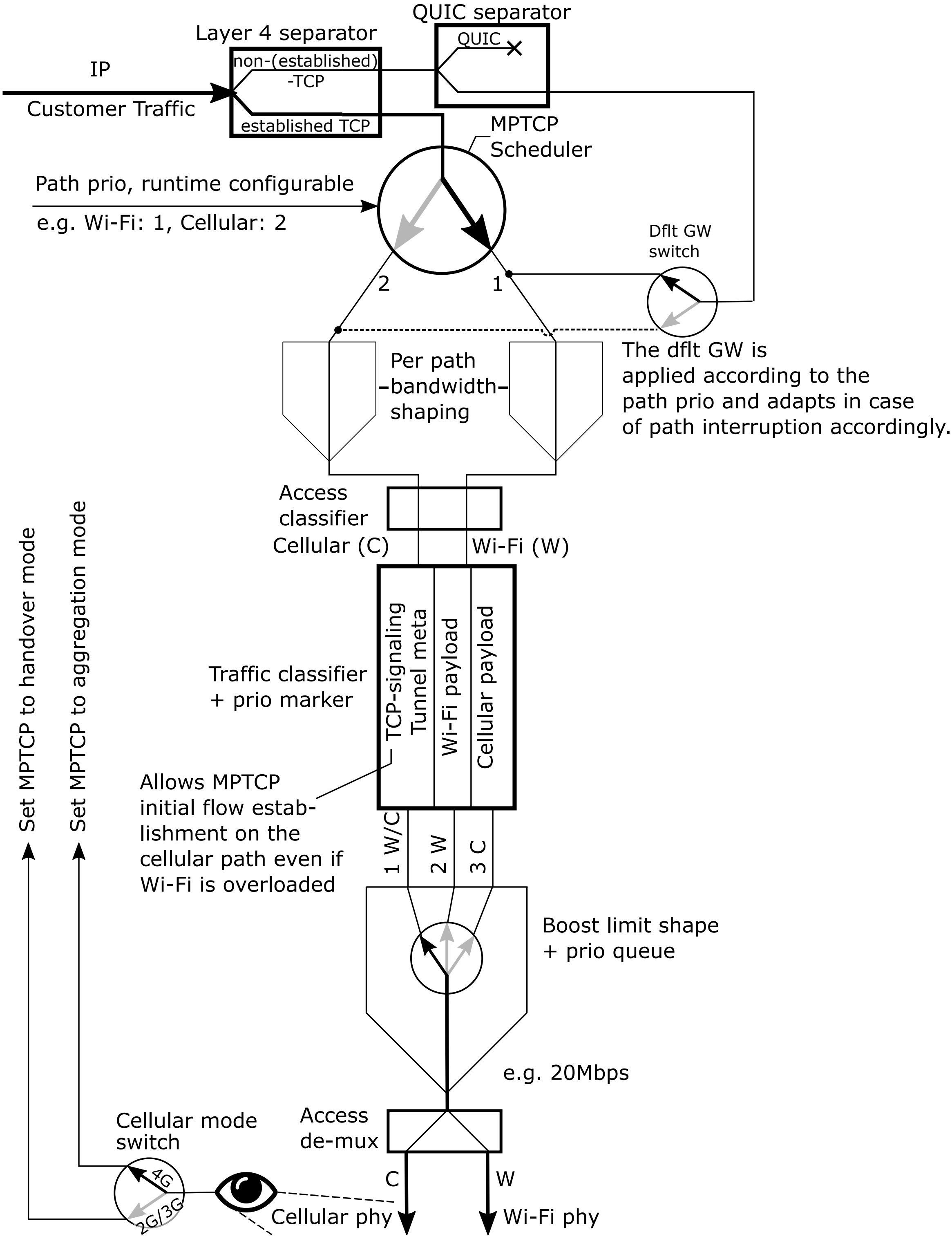


Fig. 5.21 Packet flow and QoS concept implemented for the nomadic customer trial using MPTCP

With 10 randomly selected users involved the special focus was on the verification of the COM parameter set which was also used to demonstrate the effect over VoD in section 5.3 and section 5.4 and website calling in section 5.5. In Table 5.2 these are summarised again and were applied over the 1-month test with the users.

Variable	Value
$T_{GAPthresh}$	600 ms
$T_{notraffic}$	50 ms
V_{max}	100 pkts
T_{Delay}	4 s
$T_{StartDelay}$	4 s

Table 5.2 COM parameters used for nomadic user trial

The clear objective of this exercise is, on the one hand, to verify the positive cost effect found in the testbed measurements by measuring the costs and, on the other hand, to confirm that COM can be used universally instead of CPF. The latter is all the more important because it is simply not possible to test all service variations in the testbed.

It is known that the selection of 10 users, even if randomly selected, does not give a complete picture and is not representative from a statistical point of view, but it may be a positive indication to continue the evaluation of COM beyond this work, or it may indicate a non-optimal set of parameters or the need to limit the scope of COM to other types of services/traffic, or that the idea behind COM is not feasible at all. Overall, the 10-user evaluation is a great opportunity for this work to analyse COM in the wild for the purposes of ATSSS and HA. Not to be underestimated is the effort it takes to design, prepare, implement, maintain and analyse a 4-month trial, as well as the financial and time effort involved to buy equipment and the proxy resources and to significantly modify the network stack and user interface. Even greater experiments are only conceivable if such frameworks as ATSSS are available in commercial devices without the need for their own implementation.

During the experiment, a number of parameters were recorded to get a picture of how COM works but also, in a larger context, to understand how MPTCP can be used in a scalable way. Although the following Table 5.3 lists all the parameters that were recorded, only those highlighted in bold will be discussed in this work.

During the course of this trial, several 100 GB of raw data were generated across a large number of databases collected on the proxy. After the trial, this data was consolidated into what was eventually a 190 MB database, which serves as the basis for interactively generating different types of graphs with filters for devices, traffic direction, time periods/phases and

Parameter(s)	Resolution	Unit	Possible values
Volume per access and direction	1 s	Byte	
Access type	1 s	INT	1 - Wi-Fi 2 - 2G 3 - 3G 4 - 4G
Wi-Fi cluster	1 s	INT	1 - At Home 2 - Nomadic 3 - Partner Hotspot
Per access availability	1 s	Bool	True False
TCP socket states	1 s	INT	LISTEN SYN-SENT SYN- RECEIVED ESTABLISHED FIN-WAIT-1 FIN-WAIT-2 CLOSE-WAIT CLOSING LAST-ACK TIME-WAIT CLOSED
Number of TCP sockets	1 s	INT	
Number of blocked QUIC connections	1 s	INT	
Number of UDP connections	1 s	INT	

Table 5.3 Collected performance parameters of the nomadic field trial per user device

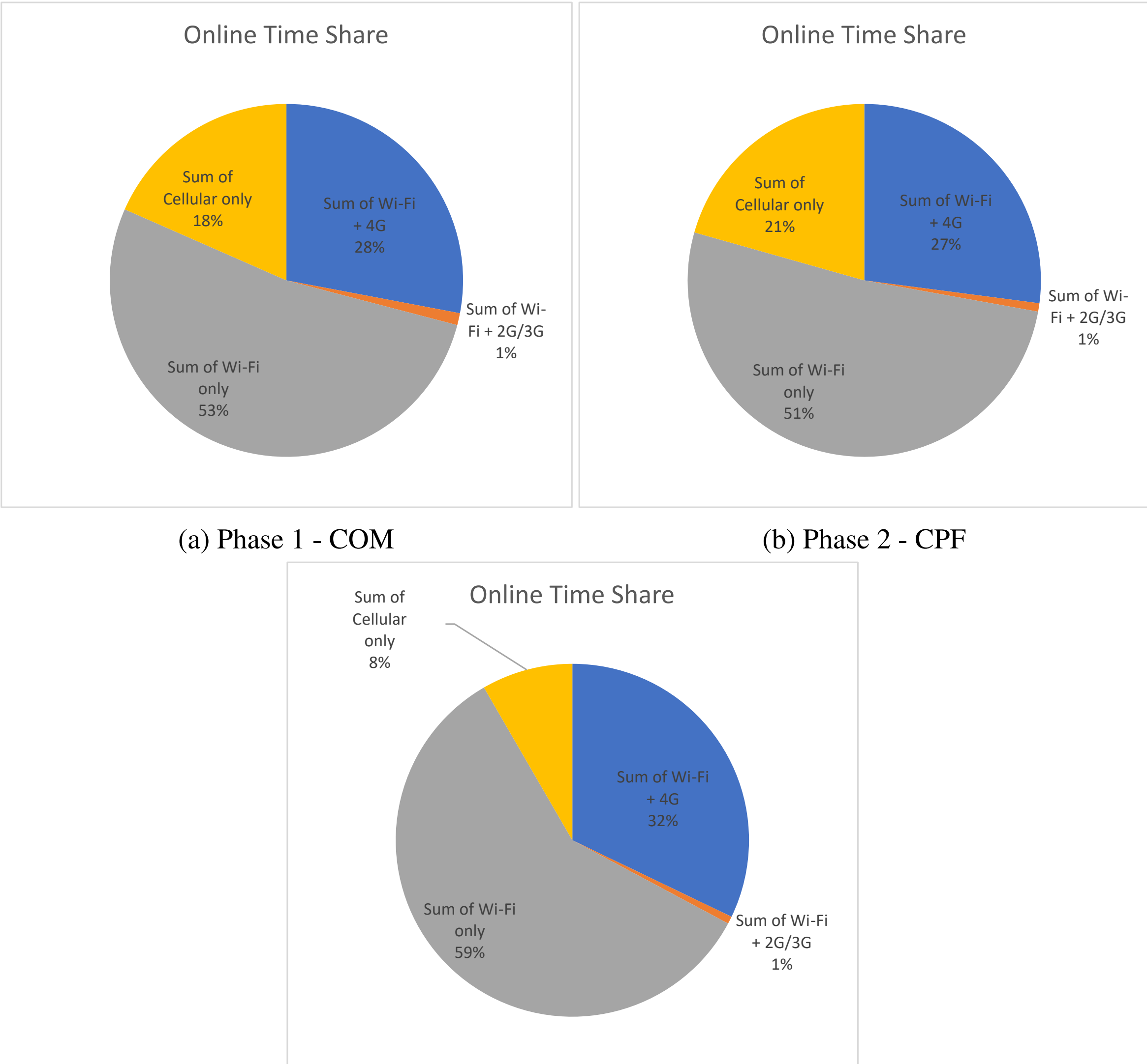
Wi-Fi clusters.

In the following, the technical measurements are discussed with a focus on the dominant downlink traffic (Internet to smartphone) in order to be able to correctly classify the results on cost distribution. In Figure 5.22, the percentage distribution over time of the possible access combinations is shown separately by phase. Across all phases, the distribution is fairly constant, with Wi-Fi only being the dominant form of access. The area which is of particular interest for this work, because it is the only one where MPTCP scheduling is active is the combination of Wi-Fi and LTE, moves between a quarter or a third share. At first glance, this could be interpreted to mean that multi-connectivity with the aim of splitting/bundling traffic is not very relevant and diminishes the importance of this trial activity. However, there are two things that put the picture back into perspective.

First, a technical measure to improve the energy consumption of the smartphones deactivated the MPTCP-splitting mode when the display was switched off. This helped maintain the energy efficiency of devices without MPTCP and is supported by the assumption that high throughput is not required when the user is not interacting with the smartphone. This design choice is a driver of the large Wi-Fi online share.

Secondly, the traffic distribution in Figure 5.23 with absolute values and Figure 5.24 with the percentage share impressively show that in all phases the MPTCP scheduler was involved in three quarters of each traffic exchange. This is an excellent result, which on the one hand shows that with certain means the predominant availability of multi-connectivity can be ensured, and on the other hand provides a good basis to strengthen the indicative results of this trial due to only 10 users. Another information is the total amount of data carried in the different phases. The first phase – COM – caused about twice the load than in the respective phases 2 and 3. While there is no clear explanation for this, it did put the COM scheduler under high workload and increases the significance of the cost split investigation.

For the purpose of considering the impact on costs, the traffic that was exchanged when the MPTCP was involved is primarily relevant, and therefore refers to the blue parts of the diagram in Figure 5.23 and 5.24. This is the subject of the Figure 5.25 with absolute values and Figure 5.26 with percentages which show the traffic distribution between LTE (4G) and Wi-Fi in this particular area. In subsection 3.2.4 it was clearly shown that the MPTCP default scheduler, which prefers the path with the lowest RTT - *minRTT* - is not able to obtain a given path cost metric. As expected, this is also visible in Figure 5.26c, which has the highest use of LTE in all three phases, while CPF in Figure 5.26b improves this by reducing the LTE share by a quarter, which is finally surpassed by COM in Figure 5.26a, which improves



(a) Phase 1 - COM

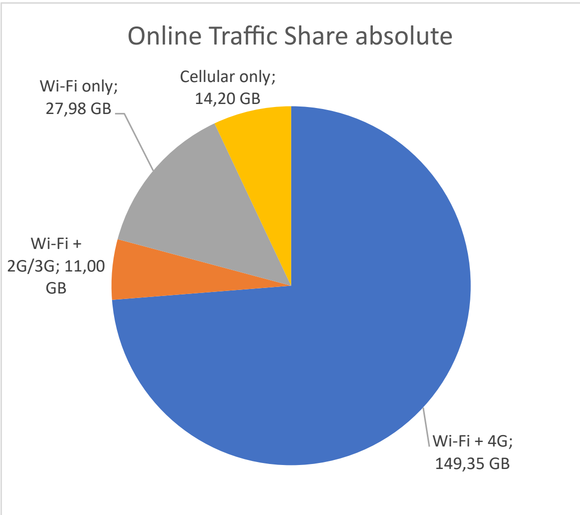
(b) Phase 2 - CPF

(c) Phase 3 - minRTT

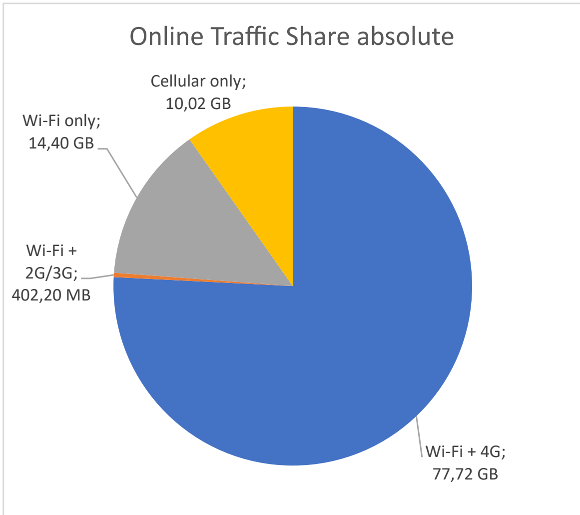
Fig. 5.22 Time share per access combination across all trial participants, separated according to the phases

by more than half compared to *minRTT* and by more than a third compared to CPF. The results are also not shown differently when looking at the share between access types in general, including the phases with only Wi-Fi and only cellular in Figure 5.27 or the same, but excluding the share of only cellular Figure 5.28.

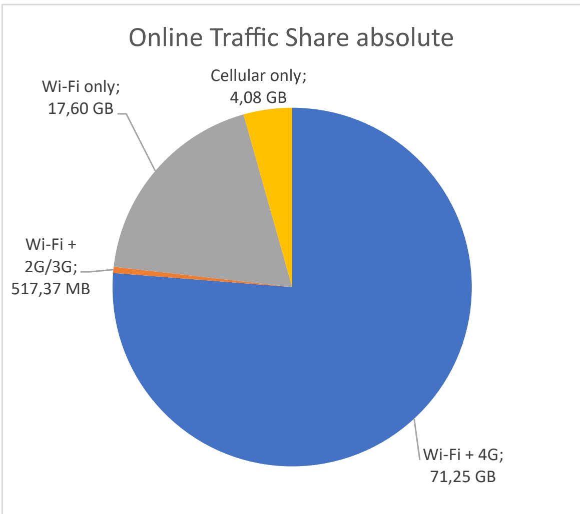
This clearly underpins the cost measurements made so far in the testbed environment and gives a more realistic picture of what can be achieved in a nomadic ATSSS use case with a traffic mix that is not just VoD and where the cheapest path bandwidth varies. Furthermore,



(a) Phase 1 - COM

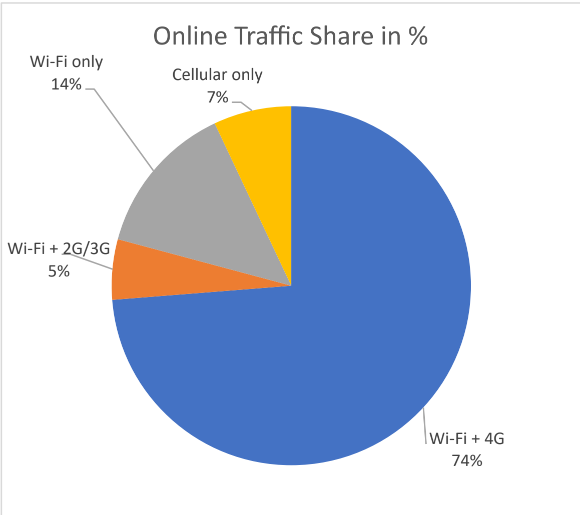


(b) Phase 2 - CPF

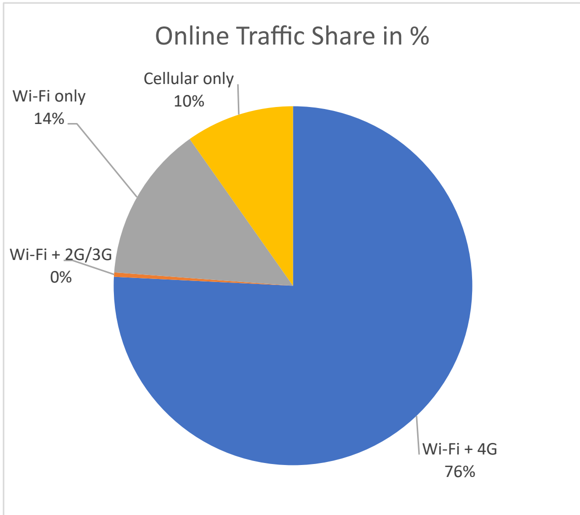


(c) Phase 3 - minRTT

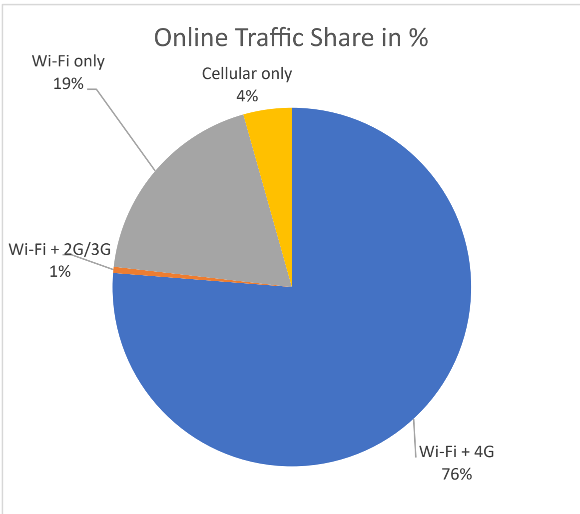
Fig. 5.23 Absolute distribution of traffic per access combination across all trial participants, separated according to the phases



(a) Phase 1 - COM

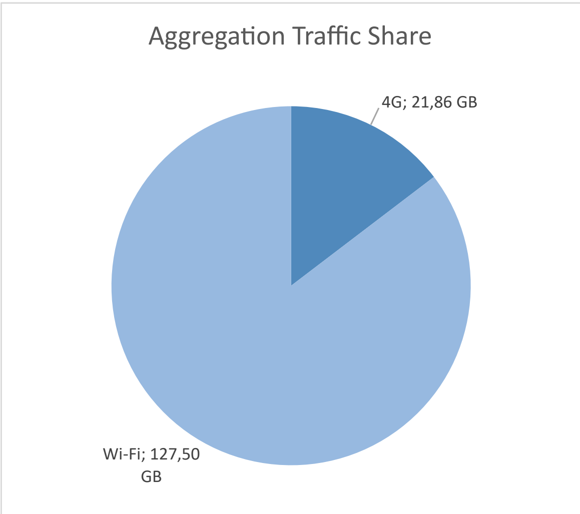


(b) Phase 2 - CPF

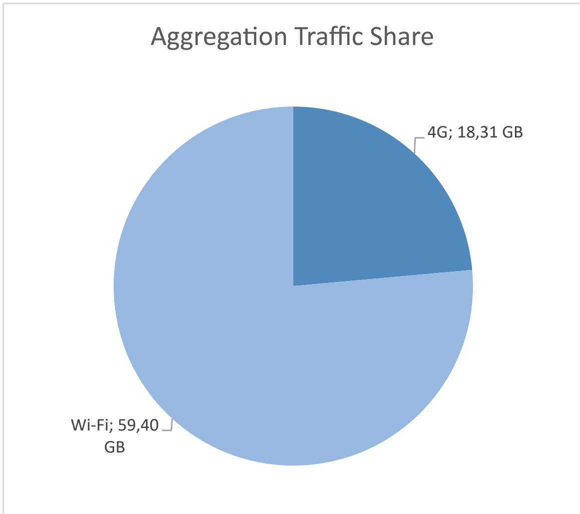


(c) Phase 3 - minRTT

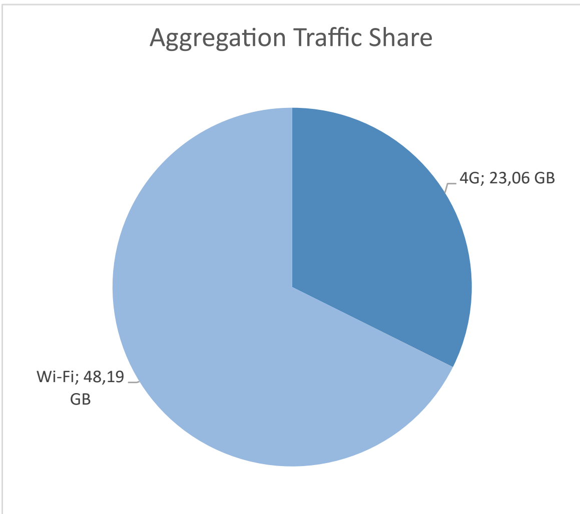
Fig. 5.24 Percentage distribution of traffic per access combination across all trial participants, separated according to the phases



(a) Phase 1 - COM

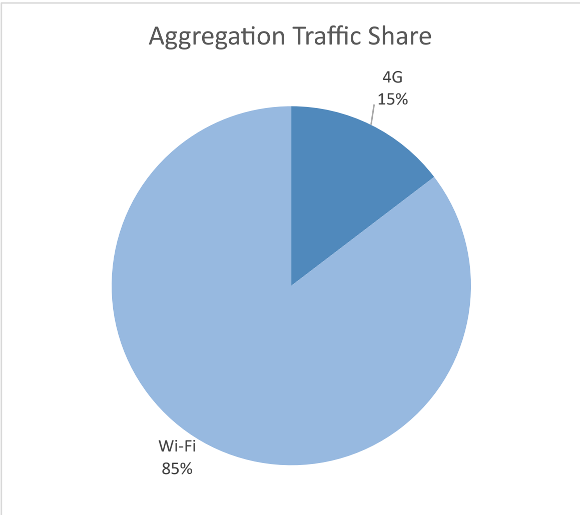


(b) Phase 2 - CPF

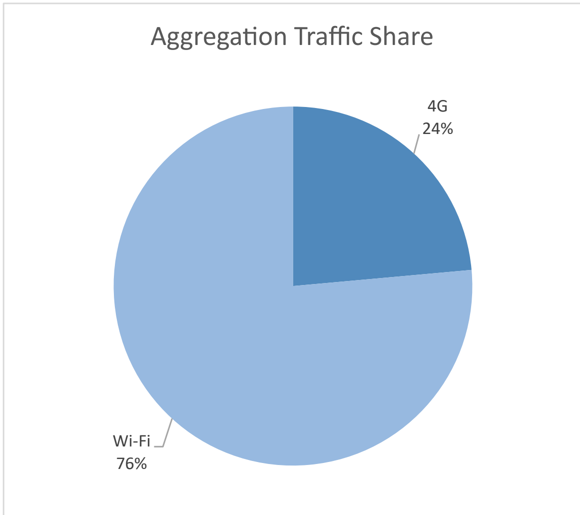


(c) Phase 3 - minRTT

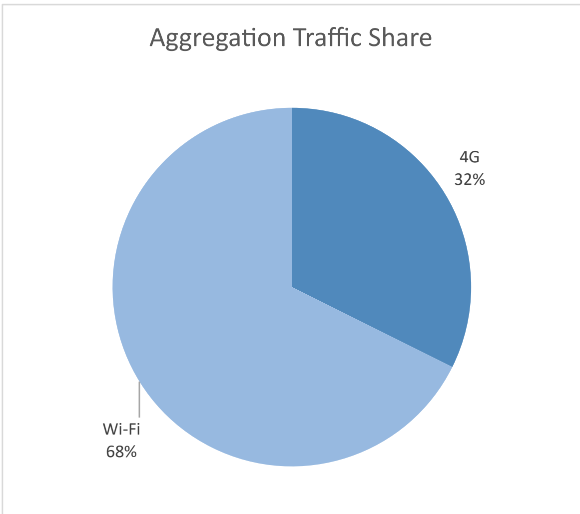
Fig. 5.25 Absolute distribution of data traffic across all trial participants if the combination of Wi-Fi and LTE allows MPTCP traffic to be split, separated according to the phases



(a) Phase 1 - COM

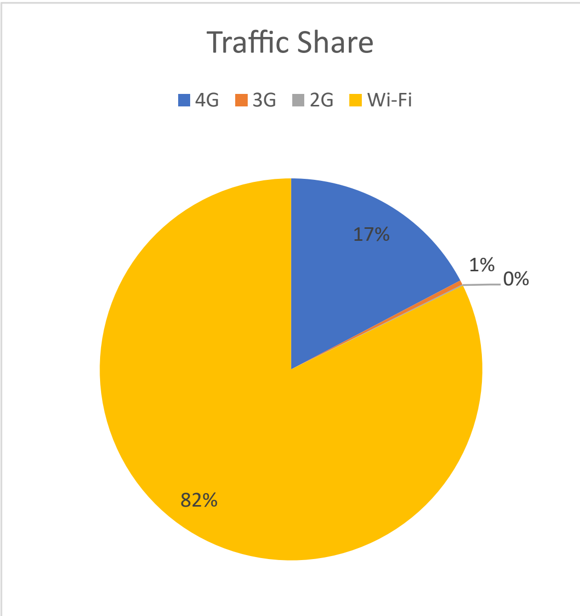


(b) Phase 2 - CPF

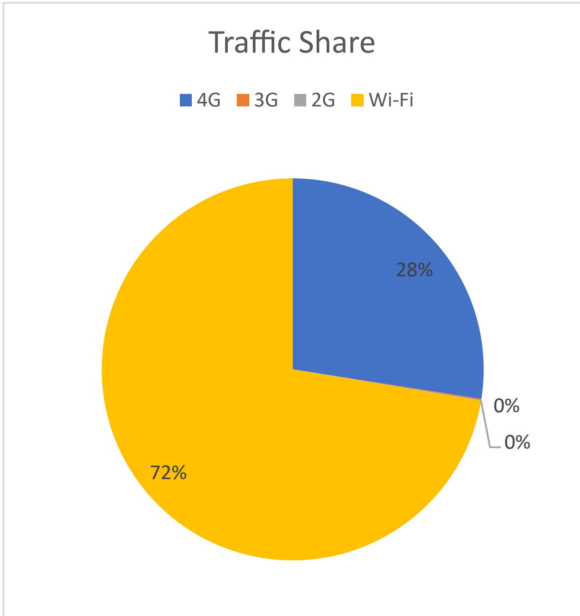


(c) Phase 3 - minRTT

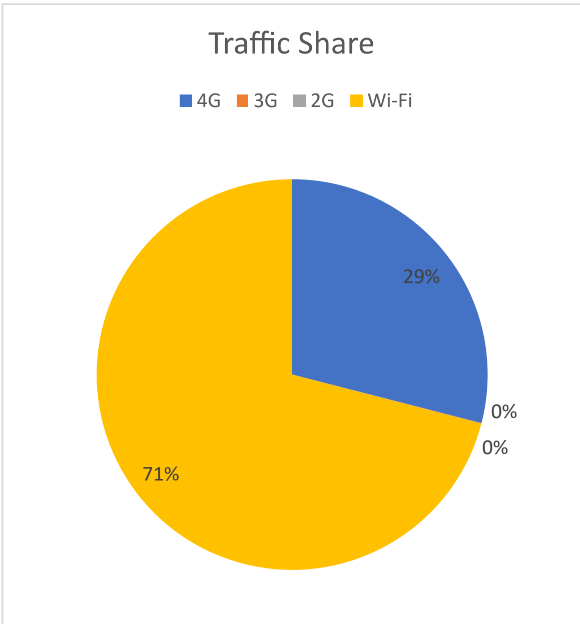
Fig. 5.26 Percentage distribution of data traffic across all trial participants if the combination of Wi-Fi and LTE allows MPTCP traffic to be split, separated according to the phases



(a) Phase 1 - COM

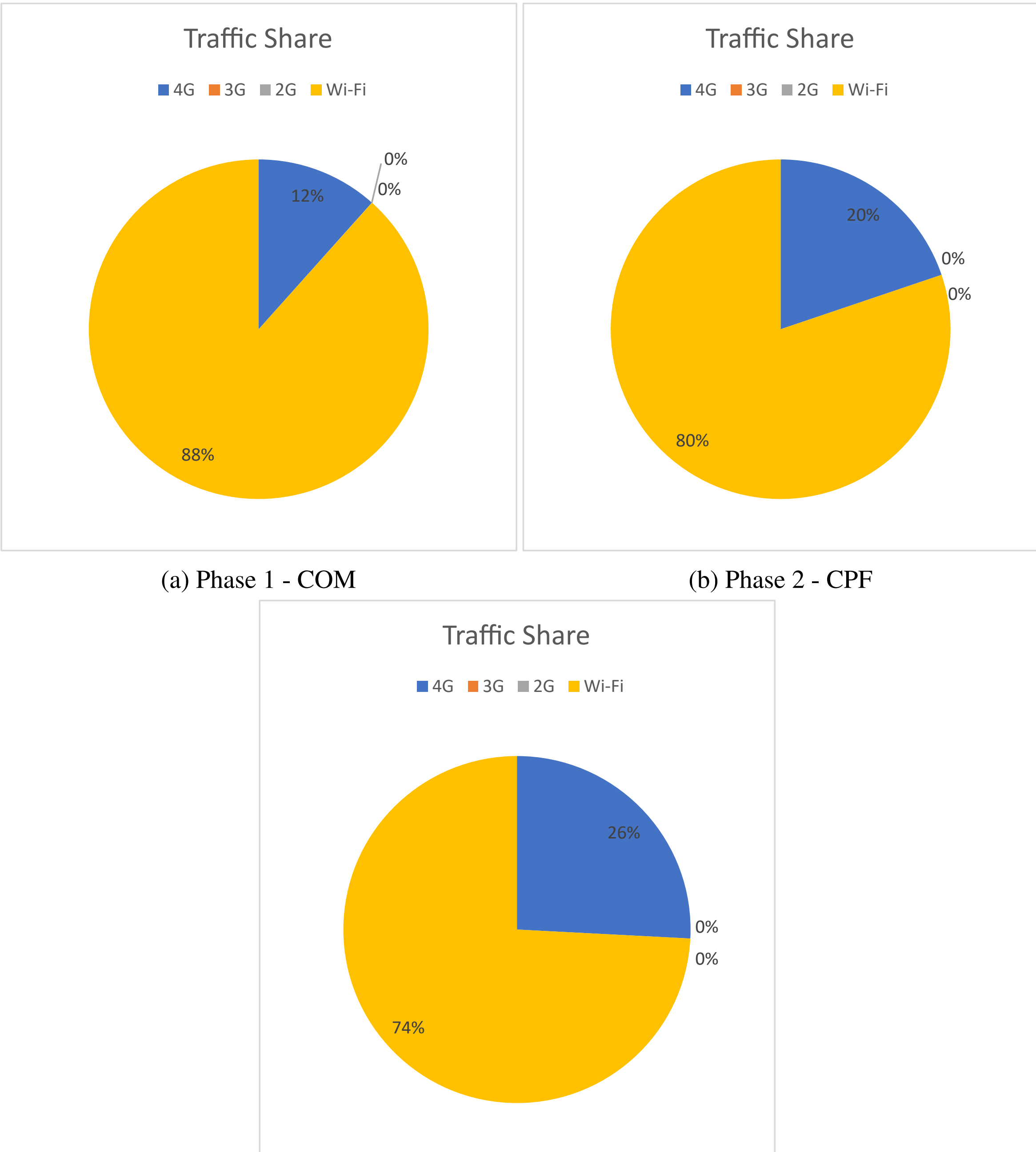


(b) Phase 2 - CPF



(c) Phase 3 - minRTT

Fig. 5.27 Percentage distribution of traffic per access across all trial participants, separated according to the phases



(a) Phase 1 - COM

(b) Phase 2 - CPF

(c) Phase 3 - minRTT

Fig. 5.28 Percentage distribution of traffic per access across all trial participants without cellular only traffic, separated according to the phases these are also the results of only one COM parameter set, which has been shown to perform well in testbed experiments, but may not be fully optimised for the nomadic scenario.

It can be also expected that in a Hybrid Access (HA) scenario the cost reduction potential can be significantly increased with the so far used COM parameter set when the cheapest path bandwidth is stable and beyond 3-6 Mbps and VoD consumption is the main traffic source.

What is missing now is the assessment of QoE, which cannot technically be measured due to an arbitrary traffic mix that does not allow the principles used in the testbeds with VoD and website to be applied. The solution of regular surveys helped to understand the impact on QoE, with the main weekly question being whether the user found any deficiencies in daily use. What was not transparent to the users at the time of the survey was the monthly transition from the acclimatisation phase to the multi-connectivity phase with the monthly change of planners. While there was a significant positive deviation from QoE when the acclimatisation phase ended and the multi-connectivity phase began, there was no longer any deviation within the multi-connectivity phase.

5.7 Analysis

After a initial parameter set was determined in section 5.1, tests with Youtube static (section 5.3) and adaptive video playout (section 5.4), Website calling (section 5.5) and a trial with mobile customer of a Tier 1 operator (section 5.6) revealed substantial good results across a range of practical relevant scenarios.

In the first evaluation step, an initial parameter set for COM was determined. This parameter set for COM consists of $t_{GAPthresh} = 600\,\text{ms}$, $t_{notraffic} = 50\,\text{ms}$ and $V_{max} = 100\,\text{pkts}$, and it turned out that $T_{Delay} = T_{StartDelay}$ is the parameter with the greatest impact to steer the operating point between the two dimensions QoE and cost. Especially when the low-cost path provides lower throughput, the intersection of cost and QoE is subject to prioritising one of both. Towards the higher low-cost path throughput, QoE is consistent and cost is only impacted by COM as outlined above.

Looking into the the results of Youtube VoD streaming with static and adaptive video resolution a first general statement can be made: So far, the visual COM results and the discussed QoE gain(s) γ and the cost factor $\Delta\theta$ with Youtube VoD streaming **always** show a QoE benefit compared to the single path transmission, while the costly path consumption is **always** reduced compared to CPF, **in most cases significantly**. Furthermore, COM can **always** provide the same QoE level as CPF when the low-cost path throughput goes beyond 6 Mbps, while minimizing and most often eliminating CPF's always present spurious demand on the high-cost path (in the range up to 100 Mbps DSL). At the lower boundary, below 6 Mbps on the low-cost path, a fine granular tuning of the COM algorithm is required when QoE is not acceptable.

A T_{Delay} of 4 s seems, however, to provide over all tests a good trade-off between the costly traffic share reduction and the perceived QoE. For non-volatile multipath systems such as Hybrid Access (HA), this value might be easily adapted to the optimal T_{Delay} for a certain DSL (low-cost path) throughput according to the presented results. Using a $t_{GAPthresh}$ of 600 ms seems to be a good choice (no results generated for other values so far). Using different CCs changes the landscape in terms of maximum efficiency, but the general statement from above is not invalidated. The differences in results can be explained by the individual calculation of the send window on the cheaper path and to what extent the send window can satisfy the burst demand. The investigated representative range of the DSL throughput from 1 to 100 Mbps shows in any case a cost optimization by COM. Even at high DSL throughput such as 50 Mbps or 100 Mbps, where one might think that VoD burst can be completely covered by DSL, COM eliminates significant spurious demand in the range of 30-40 %.

Similar tests were executed to verify COMs efficiency with video resolutions different to 1080p. It was noted that lower video resolution typically means lower throughput demand and higher video-resolution means higher throughput demand. For the use of COM, this can be reduced to a simple formula: Maximum efficiency comes with highest video resolution.

Also the verification of COM with the same identified set of parameters in a nomadic user trial with 10 users showed a significant reduction of cost compared to CPF and also proved again that the selection of the MPTCP default scheduler which prefers low RTT paths is not respecting path costs. Although this scenario, which is similar to ATSSS, is challenging due to an arbitrary traffic mix and a volatile, cheap Wi-Fi path, one third of the cost of CPF could be saved without compromising QoE in general.

Overall, this seems to be a clear indication that COM can generally replace CPF in many HA and ATSSS scenarios with a significant cost advantage and no noticeable impact on QoE, and it is likely that similar results can be achieved with MPTCP end-to-end.

Chapter 6

Conclusion

This work addresses a real-world problem in multipath transmission networks when multipath scheduling is determined solely by the access costs (such as in today's Hybrid Access (HA) or in an appropriately configured ATSSS with the priority-based steering mode), but bursty traffic undermines this by leading to unexpected transmission costs. In particular, this applies to the VoD services, where spurious demand is measured on the expensive access without resulting in better QoE compared to using only the cheaper path. A detailed analysis of this problem identified the peaks of the traffic bursts to cause this unnecessary overflow. This led to the definition of design goals for a robust cost-based scheduling algorithm which can address this problem. This work presents a new algorithm called Cost Optimized Multipath (COM) which will simultaneously reduce the cost of multipath use for network operators and also retain the QoE levels required by the end-users. This is achieved by a simple change to the design of the so far used Cheapest-path-first (CPF) algorithm for access prioritization that takes into account the time gap between packets in addition to the access costs. COM is designed to be a service agnostic approach to optimize transmission cost when dumb cost based multipath scheduler fails. It conforms to the specification of the *Priority based* steering function for the splitting of traffic within the 5G ATSSS multipath framework.

This document follows a clear structure from the motivation of this work with results from the largest commercial HA deployment demonstrating the need to counteract excessive costs in order to successfully verify a solution in the field in the even more challenging multipath environment of nomadic ATSSS for usage with smartphone. This suggests a high practical relevance as more of these deployments emerge. While HA is already established in the market at operators like Deutsche Telekom [8], Proximus [133], KPN [134], Telia [135], BT [136] and others, first ATSSS availability announcements join the stage [137].

The research conducted in this thesis addresses research questions Q1-Q5 from section 1.2. The detailed answers can be found in the following responses A1-A5.

A1 **Are the observed spurious requests for expensive path resources in HA and ATSSS multi-connectivity networks with prioritization of cheap path resources provable and traceable to specific patterns?**

As indicated by observations in section 1.1, VoD traffic regularly and significantly uses expensive cellular resources in a HA multipath connection composed of DSL and LTE. This suspicion was further substantiated in a test environment in subsection 3.2.5 that implements the dumb Cheapest-path-first (CPF) principle used in HA in a MPTCP environment, which gives strict priority to the cheaper path and overflows to the expensive path when exceeded. In a typical HA combination of 16 Mbps DSL and LTE a 1080p video consumed 90 % of cellular resources. Most importantly, however, is the realization that the QoE does not take advantage of the fact that expensive cellular access contributes significantly to the transmission of the video since the playback was smooth with and without cellular access. Finally, the evaluation in chapter 5 of the counteracting COM scheduler solution developed in this work compared to CPF also proves that in a wide range of HA and ATSSS scenarios with typical DSL or Wi-Fi bandwidths, CPF is prone to cost inefficiencies.

Following the analysis of the transmission principle of VoD in section 2.4 and the confidence gained in the measurement in subsection 3.2.5 that the cheap path is not utilized, the realization matured: Services, in particular VoD, that transmit their data as fast as possible, tend to consume the available transmission channel, even if it is an aggregated channel of a multipath system. When the capacity of the transmission channel is higher than the average output rate of the service, a traffic pattern of a burst or successive bursts with peaks spilling over into the expensive path in HA or ATSSS is created. This is particularly the case for services such as VoD where chunks of data are transmitted sequentially. In such a case when the expensive path is used, while the cheap path is not utilized spurious expensive demand is created.

A2 **In the landscape of existing multi-path solutions, are there answers to the question of how to curb the occurrence of spurious expensive requests?**

An analysis of the genesis of multipath concepts in section 2.1 and a literature research on multipath traffic scheduler did not identify any solution which can solve the conflict of VoD with its bursty traffic pattern and cost efficient multipath scheduling in the scope of a service transparent multipath framework as provided with HA and ATSSS.

Two analyzed candidates [92] and [93], which provide a solution space for cost-efficient multipath transport of VoD, require the scheduler built into the VoD service or access to the video coding information. While this is not an acceptable solution for a service transparent multipath framework, these two solutions point out the – for this thesis – important aspect of *delay-tolerance* of VoD when a reasonable playout buffer at the client can cache video data which is received with a faster rate than the playback rate.

Thus, the fundamental problem is not a new one but remains unconsidered in the specification of scheduling approaches of HA and ATSSS as outlined in section 2.2 and section 3.8. Although both approaches specify a cost prioritized scheduling of access paths, the detailed implementation is left open.

A3 **What is a viable technical solution that does not require interaction with the service, that promises an efficient way to avoid unwanted expensive requests without decreasing QoE?**

The main purpose of the multipath frameworks HA and ATSSS is to increase the QoE by aggregating cellular access to the fixed access or the Wi-Fi. The basic question is, however, how much multipath aggregation of resources it requires to achieve the goal of better QoE. This describes exactly the two dimensions which need to be balanced more efficiently as it is the case with the dumb CPF scheduler.

This realisation is linked in section 3.5 to the response of A2, which identifies the bursty traffic pattern of VoD as the cause and provides a solution that depends solely on the analysis of the traffic pattern. Under regular conditions, the available bandwidth for the transport of the video data is the only parameter which determines about the QoE as figured out in section 4.2. In the case where the playback rate of the video data is equal to the available bandwidth, the most efficient case in terms of cost and QoE is achieved and basically means a constant use of the transmission channel without gaps. However, the problematic case observed in A1's response is the generation of a burst-like traffic pattern and a resulting non-constant use of the transmission channel due to gaps between bursts. The developed solution idea, called Cost Optimized Multi-path (COM), is designed to cut off the tops of the traffic peaks when gaps are measured during a transmission. This is achieved by blocking the expensive path for a specified time in the scheduling process and is not applied if a gap cannot be measured to avoid a QoE deterioration.

A4 **What are the barriers to integrating such a technical solution?**

The design principles formulated in subsection 3.5.2 provide a low barrier to integration of the basic COM idea, which is then demonstrated in section 3.6 with an implementation in the standardised and open-source multipath protocol MPTCP, which is also used for multipath transport in ATSSS. COM extends the CPF logic with a gap detection logic based on simply measuring the time between packets and indicating based on this the availability of the expensive path in the scheduling process. Compared to an implementation of CPF in MPTCP this does not add a performance overhead as it reuses existing TCP functions or makes use of simple arithmetic which is not recognized.

Since the design principles enforce efficient integration with service-transparent multipath systems, the remaining hurdle is to understand how to parameterise COM to achieve the expected cost reduction results within the delay tolerance of VoD addressed in A2.

A5 **Is the implemented solution superior to the current approach of dumb cost prioritization of access paths?**

Yes. In all verified scenarios with VoD and other services in different environments, reduction or elimination of expensive path costs without impacting QoE was possible, depending on the selected COM parameter set.

The main factors that could be identified that determine the efficiency of COM are the bandwidth of the cheaper path and the COM parameters responsible for the gap detection $t_{GAPthresh}$ and the resulting blocking of the expensive path for a certain time T_{Delay}.

A generic parameter set found independent of the bandwidth of the cheap path showed significant cost reduction with some impact on QoE compared to CPF when the bandwidth of the cheap path is quite low. On the other hand, COM holds the promise of multipath to deliver better QoE than singlepath. In a ATSSS modeled trial with mobile customers of a Tier 1 MNO, the generic parameter set was found to be the most cost-effective compared to CPF and *minRTT*, while QoE was found to be stable.

The answers to the research questions also answer the PhD objectives, as these are derived from them. Therefore, the results of this work suggest under specified circumstances a risk-free use of COM as a substitute for CPF, whereby the confidence can actually only be increased by expanding the testbeds presented here to include more usage scenarios and completing intensive measurement campaigns, or by testing COM in scaling commercial deployments.

6.1 Future work

The analysis in this work has shown that COM is a viable approach to counteract the treatment of cost-problematic traffic patterns where CPF fails. With the different parameters of COM the balance between cost and QoE can be fine granular controlled as it is shown in the broad range of results of this work. On the other hand, a generic parameter set was able to deliver in daily usage trials convincing first result over the status quo with CPF without fine tuning for certain scenarios. Although the results of this work enable an immediate benefit, if COM is used in a MPTCP multipath framework such as ATSSS, there remains some room for optimisation and enhanced usage scenarios as analysed below as possible future work.

- **Implementation in GRE based Hybrid Access (HA)**:

  The CPF principle implemented in the GRE based HA and the one implemented in MPTCP show the same impact over VoD. This suggests that the COM implemented in this work for MPTCP qualifies also for a GRE implementation. This statement also draws its plausibility from the fact that COM is based on the traffic pattern seen on the transmission path which is the same independent if this is considered on the TCP or GRE layer.
  The difference, however, is that in a GRE-based HA, a traffic mix is to be expected due to the simultaneous use of several services in a residential environment. How this affects the efficiency of COM and whether countermeasures are needed needs to be further investigated.

- **Non-monetary cost structures**:

  COM's motivation and design follows the idea of a financial cost driven prioritization of access paths in a multipath system. However, also other costs are imaginable for example latency when shortest path communication is critical for QoE. In theory, COM should do the same, as it is designed to keep traffic on the prioritised path as much as possible, but prioritised in this case due to lower latency. If this helps to keep the overall latency low if peaks are not scheduled over high latency paths needs further evidence.

- **Self-learning COM**:

  Methods like machine learning or artificial intelligence use pre-trained models to measure the impact under operation to derive optimized parameter sets. This can be used to optimize COM for various scenarios offline or online without manual testing. This increases the understanding of the optimization potential of COM and lets COM self-adjust to the best operating point in both dimensions of cost and QoE.

- **Verification of COM's design principles with other multipath protocols and in end-to-end deployments**:

  As ATSSS evolves, 3GPP is investigating other multipath protocols, namely MP-QUIC and MP-DCCP. In addition, end-to-end multipath is expected to become more tangible over time as cost pressures in cellular networks ease.
  Since the design of COM is protocol-independent and the two multipath network protocols mentioned above share similarities with MPTCP, e.g., scheduling using CC information, suggests a straightforward adoption from the COM MPTCP implementation described in detail in this thesis.
  In addition, the design principles does not exclude the implementation of COM in an end-to-end multipath deployment. While it is true that the design principles require integration into a service transparent multipath framework, this may well be seen end to end.

- **Identification and handling of paths in a multipath system that contribute negatively to the QoE of VoD**:

  The definition of the QoE dependency of VoD in section 4.2 requires the exclusion of paths which cannot be covered by the client receive buffer used for VoD playback. Two things which are likely in this context but need further evidence are:

  1. This is a general multipath issue when a path in a multipath system has an impact to the overall performance and/or impacts the service.
  2. Within HA this is probably less of a problem since the operator typically controls both access path and can therefore ensure non-problematic path characteristics.

It is subject to further research to analyse how severe this requirement is in relation to COM and ATSSS and how problematic paths can be identified and excluded from the scheduling process. For example, [91] with its ability to identify the contribution of a path to the overall goal of high throughput could be a good starting point.

References

[1] N. Keukeleire, B. Hesmans, and O. Bonaventure, "Increasing Broadband Reach with Hybrid Access Networks", *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 43–49, Mar. 2020, ISSN: 2471-2833. DOI: 10.1109/MCOMSTD.001.1900036.

[2] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "Observing real smartphone applications over multipath TCP", *IEEE Communications Magazine*, vol. 54, no. 3, pp. 88–93, Mar. 2016, ISSN: 1558-1896. DOI: 10.1109/MCOM.2016.7432153.

[3] M. Condoluci *et al.*, "Fixed-Mobile Convergence in the 5G Era: From Hybrid Access to Converged Core", *IEEE Network*, vol. 33, no. 2, pp. 138–145, Mar. 2019, ISSN: 1558-156X. DOI: 10.1109/MNET.2018.1700462.

[4] J. Lee, Y. Yi, S. Chong, and Y. Jin, "Economics of WiFi Offloading: Trading Delay for Cellular Capacity", *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1540–1554, Mar. 2014, ISSN: 1558-2248. DOI: 10.1109/TWC.2014.010214.130949.

[5] Y. He, M. Chen, B. Ge, and M. Guizani, "On WiFi Offloading in Heterogeneous Networks: Various Incentives and Trade-Off Strategies", *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2345–2385, 2016, ISSN: 1553-877X. DOI: 10.1109/COMST.2016.2558191.

[6] L. Kim, *5G Economics – The Numbers*, https://techneconomyblog.com/2017/07/, Jul. 2017.

[7] L. Kim, *Fixed Wireless Access in a Modern 5G Setting – What Does it Bring That We Don't Already Have?*, https://techneconomyblog.com/2023/01/, Jan. 2023.

[8] N. Leymann, "Hybrid Access deployment @ DT", Internet Engineering Task Force, Tech. Rep., Nov. 2016. [Online]. Available: https://datatracker.ietf.org/meeting/97/materials/slides-97-banana-hybrid-access-deployment-at-deutsche-telekom-01.pdf.

[9] N. Leymann, C. Heidemann, M. Zhang, B. Sarikaya, and M. Cullen, *Huawei's GRE Tunnel Bonding Protocol*, RFC 8157, May 2017. DOI: 10.17487/RFC8157. [Online]. Available: https://www.rfc-editor.org/info/rfc8157.

[10] A. Ford, C. Raiciu, M. J. Handley, O. Bonaventure, and C. Paasch, *TCP Extensions for Multipath Operation with Multiple Addresses*, RFC 8684, Mar. 2020. DOI: 10.17487/RFC8684. [Online]. Available: https://www.rfc-editor.org/info/rfc8684.

[11] 3GPP, "System architecture for the 5G System (5GS)", 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, Dec. 2021, Version 16.11.0.

[12] BBF, “5G Wireless Wireline Convergence Architecture”, Broadband Forum, Technical Report (TR) 470, Aug. 2020, Issue 1.

[13] M. Amend, A. Brunstrom, A. Kassler, V. Rakocevic, and S. Johnson, “DCCP Extensions for Multipath Operation with Multiple Addresses”, Internet Engineering Task Force, Internet-Draft draft-ietf-tsvwg-multipath-dccp-13, Jan. 2024, Work in Progress, 52 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tsvwg-multipath-dccp/13/.

[14] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, “Multipath Extension for QUIC”, Internet Engineering Task Force, Internet-Draft draft-ietf-quic-multipath-03, Oct. 2022, Work in Progress, 34 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/03/.

[15] 3GPP, “Study on access traffic steering, switching and splitting support in the 5G system architecture; Phase 3”, 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.700-53, Apr. 2022, Version 0.2.0.

[16] O. Bonaventure, M. Piraux, Q. D. Coninck, M. Baerts, C. Paasch, and M. Amend, “Multipath schedulers”, Internet Engineering Task Force, Internet-Draft draft-bonaventure-iccrg-schedulers-02, Oct. 2021, Work in Progress, 14 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-bonaventure-iccrg-schedulers/02/.

[17] Cisco, “Cisco Visual Networking Index: Forecast and Methodology, 2016–2021”, Cisco, Tech. Rep., Jun. 2017.

[18] Sandvine, “The Mobile Internet Phenomena Report, May 2021”, Sandvine, Tech. Rep., May 2021.

[19] Ericsson, “Ericsson Mobility Report November 2022”, Ericsson, Tech. Rep., Nov. 2022.

[20] Deutsche Telekom. “DEUTSCHE TELEKOM – Annual Report 2019”, Deutsche Telekom. (Oct. 7, 2023), [Online]. Available: https://report.telekom.com/annual-report-2019/management-report/development-of-business-in-the-operating-segments/germany.html (visited on 2019).

[21] M. Amend, V. Rakocevic, and J. Habermann, “Cost optimized multipath scheduling in 5G for Video-on-Demand traffic”, in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2021, pp. 1–6. DOI: 10.1109/WCNC49053.2021.9417415.

[22] M. Amend and V. Rakocevic, “Cost-efficient multipath scheduling of video-on-demand traffic for the 5G ATSSS splitting function”, *Computer Networks*, vol. 242, p. 110 218, 2024, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2024.110218. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128624000501.

[23] M. Pieska, A. Rabitsch, A. Brunstrom, A. Kassler, and M. Amend, “Adaptive Cheapest Path First Scheduling in a Transport-Layer Multi-Path Tunnel Context”, in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW ’21, Virtual Event, USA: Association for Computing Machinery, 2021, pp. 39–45, ISBN: 9781450386180. DOI: 10.1145/3472305.3472316. [Online]. Available: https://doi.org/10.1145/3472305.3472316.

[24] M. Pieska, A. Rabitsch, A. Brunstrom, A. Kassler, M. Amend, and E. Bogenfeld, "Low-delay cost-aware multipath scheduling over dynamic links for access traffic steering, switching, and splitting", *Computer Networks*, vol. 241, p. 110 186, 2024, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2024.110186. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128624000185.

[25] C. Lange *et al.*, "Bridging the Last Mile", in *Broadband Coverage in Germany; 10. ITG-Symposium*, Apr. 2016, pp. 1–8.

[26] N. Bayer, A. T. Girmazion, M. Amend, K. Haensge, R. Szczepanski, and M. D. Hailemichael, "Bundling of DSL resources in home environments", in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016, pp. 1–7. DOI: 10.1109/WoWMoM.2016.7523515.

[27] M. Amend, V. Rakocevic, A. P. Matz, and E. Bogenfeld, "RobE: Robust Connection Establishment for Multipath TCP", in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW '18, Montreal, QC, Canada: Association for Computing Machinery, 2018, pp. 76–82, ISBN: 9781450355858. DOI: 10.1145/3232755.3232762. [Online]. Available: https://doi.org/10.1145/3232755.3232762.

[28] M. Amend *et al.*, "A Framework for Multiaccess Support for Unreliable Internet Traffic using Multipath DCCP", in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, Oct. 2019, pp. 316–323. DOI: 10.1109/LCN44214.2019.8990746.

[29] N. R. Moreno, M. Amend, A. Brunstrom, A. Kassler, and V. Rakocevic, "CCID5: An Implementation of the BBR Congestion Control Algorithm for DCCP and Its Impact over Multi-Path Scenarios", in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW '21, Virtual Event, USA: Association for Computing Machinery, 2021, pp. 52–58, ISBN: 9781450386180. DOI: 10.1145/3472305.3472322. [Online]. Available: https://doi.org/10.1145/3472305.3472322.

[30] R. Alfredsson, A. Kassler, J. Vestin, M. Pieskä, and M. Amend, "Accelerating a Transport Layer based 5G Multi-Access Proxy on SmartNIC", in *Würzburg Workshop on Next-Generation Communication Networks (WueWoWas'22) :*, Würzburg University, 2022, pp. 4–. DOI: 10.25972/OPUS-28079.

[31] M. Amend *et al.*, "In-network Support for Packet Reordering for Multiaccess Transport Layer Tunneling", in *2022 IEEE 11th IFIP International Conference on Performance Evaluation and Modeling in Wireless and Wired Networks (PEMWN)*, Nov. 2022, pp. 1–6. DOI: 10.23919/PEMWN56085.2022.9963814.

[32] F. Brisch, A. Kassler, J. Vestin, M. Pieskä, and M. Amend, "Accelerating Transport Layer Multipath Packet Scheduling for 5G-ATSSS", in *KuVS Fachgespräch - Würzburg Workshop on Modeling, Analysis and Simulation of Next-Generation Communication Networks 2023 (WueWoWAS'23)*, 2023, p. 4. DOI: 10.25972/OPUS-32205.

[33] M. Amend and E. Bogenfeld, "A Communication System for Transmitting a Transmission Control Protocol Segment Over a Communication Network Using a Multipath Transmission Control Protocol, Corresponding Method and Computer Program", European pat. 3579500B1, Nov. 17, 2021.

[34] M. Amend and E. Bogenfeld, "Techniques for Scheduling Multipath Data Traffic", European pat. 3544332B1, May 26, 2021.

[35] M. Amend and E. Bogenfeld, “Techniques for Detecting Bursty Traffic Pattern Detection and Scheduling Multipath Data Traffic”, European pat. 3796604B1, Jun. 14, 2023.

[36] M. Amend and E. Bogenfeld, “A method and communication device for transmitting multiple data streams of different communication services over a multipath transmission system”, European pat. 4042653A1, Aug. 17, 2022.

[37] M. Amend, E. Bogenfeld, A. Brunstrom, M. Pieska, and A. Kassler, “System and method for multipath transmission”, European pat. 4080836B1, Jun. 21, 2023.

[38] M. Amend, E. Bogenfeld, A. Brunstrom, M. Pieska, and A. Kassler, “System and method for multipath transmission with efficient adjustable reliability”, European pat. 4075742A1, Oct. 19, 2022.

[39] M. Amend, “Techniques for an efficient use of redundant multipath scheduling”, European pat. 4044526A1, Aug. 17, 2022.

[40] M. Amend, E. Bogenfeld, A. Brunstrom, A. Kassler, and V. Rakocevic, “A multipath framework for UDP traffic over heterogeneous access networks”, Internet Engineering Task Force, Internet-Draft draft-amend-tsvwg-multipath-framework-mpdccp-01, Jul. 2019, Work in Progress, 10 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-amend-tsvwg-multipath-framework-mpdccp/01/.

[41] M. Amend, A. Brunstrom, A. Kassler, and V. Rakocevic, “Lossless and overhead free DCCP - UDP header conversion (U-DCCP)”, Internet Engineering Task Force, Internet-Draft draft-amend-tsvwg-dccp-udp-header-conversion-01, Jul. 2019, Work in Progress, 11 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-amend-tsvwg-dccp-udp-header-conversion/01/.

[42] M. Boucadair *et al.*, “3GPP Access Traffic Steering Switching and Splitting (ATSSS) - Overview for IETF Participants”, Internet Engineering Task Force, Internet-Draft draft-bonaventure-quic-atsss-overview-00, May 2020, Work in Progress, 29 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-bonaventure-quic-atsss-overview/00/.

[43] M. Amend and D. V. Hugo, “Multipath sequence maintenance”, Internet Engineering Task Force, Internet-Draft draft-amend-iccrg-multipath-reordering-03, Oct. 2021, Work in Progress, 16 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-amend-iccrg-multipath-reordering/03/.

[44] M. Amend and J. Kang, “Multipath TCP Extension for Robust Session Establishment”, Internet Engineering Task Force, Internet-Draft draft-amend-tcpm-mptcp-robe-02, Mar. 2022, Work in Progress, 28 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-amend-tcpm-mptcp-robe/02/.

[45] *IEEE Standard for Local and metropolitan area networks–Link Aggregation*, IEEE Std 802.1AX-2008, Nov. 2008, pp. 1–163. DOI: 10.1109/IEEESTD.2008.4668665.

[46] D. Carr, T. Coradetti, B. Lloyd, G. McGregor, and K. L. Sklower, *The PPP Multilink Protocol (MP)*, RFC 1990, Aug. 1996. DOI: 10.17487/RFC1990. [Online]. Available: https://www.rfc-editor.org/info/rfc1990.

[47] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2", 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.300, Jan. 2016, Version 13.2.0.

[48] R. Atkinson and S. Bhatti, *Identifier-Locator Network Protocol (ILNP) Architectural Description*, RFC 6740, Nov. 2012. DOI: 10.17487/RFC6740. [Online]. Available: https://www.rfc-editor.org/info/rfc6740.

[49] T. Herbert and P. Lapukhov, "Identifier-locator addressing for IPv6", Internet Engineering Task Force, Internet-Draft draft-herbert-intarea-ila-01, Mar. 2018, Work in Progress, 48 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-herbert-intarea-ila/01/.

[50] E. Nordmark and M. Bagnulo, *Shim6: Level 3 Multihoming Shim Protocol for IPv6*, RFC 5533, Jun. 2009. DOI: 10.17487/RFC5533. [Online]. Available: https://www.rfc-editor.org/info/rfc5533.

[51] K. Chowdhury, K. Leung, B. Patil, V. Devarapalli, and S. Gundavelli, *Proxy Mobile IPv6*, RFC 5213, Aug. 2008. DOI: 10.17487/RFC5213. [Online]. Available: https://www.rfc-editor.org/info/rfc5213.

[52] M. Menth, A. Stockmayer, and M. Schmidt, "LISP Hybrid Access", Internet Engineering Task Force, Internet-Draft draft-menth-lisp-ha-00, Jul. 2015, Work in Progress, 19 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-menth-lisp-ha/00/.

[53] S. Pierrel, P. Jokela, J. Melén, and K. Slavov, "A policy system for simultaneous multiaccess with host identity protocol", *Proc. ACNM*, pp. 71–77, 2007.

[54] A. Gurtov and T. Polishchuk, "Secure multipath transport for legacy Internet applications", in *2009 Sixth International Conference on Broadband Communications, Networks, and Systems*, Sep. 2009, pp. 1–8. DOI: 10.4108/ICST.BROADNETS2009.7186.

[55] S. Sevilla and J. J. Garcia-Luna-Aceves, "A deployable identifier-locator split architecture", in *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, Jun. 2017, pp. 1–9. DOI: 10.23919/IFIPNetworking.2017.8264833.

[56] M. Arye, E. Nordström, R. Kiefer, J. Rexford, and M. J. Freedman, "A formally-verified migration protocol for mobile, multi-homed hosts", in *2012 20th IEEE International Conference on Network Protocols (ICNP)*, Oct. 2012, pp. 1–12. DOI: 10.1109/ICNP.2012.6459961.

[57] K. Evensen, D. Kaspar, P. Engelstad, A. F. Hansen, C. Griwodz, and P. Halvorsen, "A network-layer proxy for bandwidth aggregation and reduction of IP packet reordering", in *2009 IEEE 34th Conference on Local Computer Networks*, Oct. 2009, pp. 585–592. DOI: 10.1109/LCN.2009.5355198.

[58] M. Bednarek, G. B. Kobas, M. Kühlewind, and B. Trammell, "Multipath Bonding at Layer 3", in *Proceedings of the 2016 Applied Networking Research Workshop*, ser. ANRW '16, Berlin, Germany: ACM, 2016, pp. 7–12, ISBN: 978-1-4503-4443-2. DOI: 10.1145/2959424.2959439. [Online]. Available: http://doi.acm.org/10.1145/2959424.2959439.

[59] B. Almsi, G. Lencse, and S. Szilgyi, "Investigating the Multipath Extension of the GRE in UDP Technology", *Comput. Commun.*, vol. 103, no. C, pp. 29–38, May 2017, ISSN: 0140-3664. DOI: 10.1016/j.comcom.2017.02.002. [Online]. Available: https://doi.org/10.1016/j.comcom.2017.02.002.

[60] G. Lencse, S. Szilagyi, F. Fejes, and M. Georgescu, "MPT Network Layer Multipath Library", Internet Engineering Task Force, Internet-Draft draft-lencse-tsvwg-mpt-10, Jun. 2022, Work in Progress, 29 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-lencse-tsvwg-mpt/10/.

[61] J. Iyengar, P. Amer, and R. Stewart, "Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths", *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 951–964, Oct. 2006, ISSN: 1558-2566. DOI: 10.1109/TNET.2006.882843.

[62] P. P. D. Amer *et al.*, "Load Sharing for the Stream Control Transmission Protocol (SCTP)", Internet Engineering Task Force, Internet-Draft draft-tuexen-tsvwg-sctp-multipath-23, Feb. 2022, Work in Progress, 28 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-tuexen-tsvwg-sctp-multipath/23/.

[63] K. c. Lan and C. Y. Li, "Improving TCP performance over an on-board multi-homed network", in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2012, pp. 2961–2966. DOI: 10.1109/WCNC.2012.6214311.

[64] D. Wischik, M. Handley, and M. B. Braun, "The Resource Pooling Principle", *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 5, pp. 47–52, Sep. 2008, ISSN: 0146-4833. DOI: 10.1145/1452335.1452342. [Online]. Available: https://doi.org/10.1145/1452335.1452342.

[65] Q. De Coninck and O. Bonaventure, "Multipath QUIC: Design and Evaluation", in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '17, Incheon, Republic of Korea: Association for Computing Machinery, 2017, pp. 160–166, ISBN: 9781450354226. DOI: 10.1145/3143361.3143370. [Online]. Available: https://doi.org/10.1145/3143361.3143370.

[66] V. Singh, S. Ahsan, and J. Ott, "MPRTP: Multipath Considerations for Real-Time Media", in *Proceedings of the 4th ACM Multimedia Systems Conference*, ser. MMSys '13, Oslo, Norway: Association for Computing Machinery, 2013, pp. 190–201, ISBN: 9781450318945. DOI: 10.1145/2483977.2484002. [Online]. Available: https://doi.org/10.1145/2483977.2484002.

[67] V. Singh, T. Karkkainen, J. Ott, S. Ahsan, and L. Eggert, "Multipath RTP (MPRTP)", Internet Engineering Task Force, Internet-Draft draft-ietf-avtcore-mprtp-03, Jul. 2016, Work in Progress, 41 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-avtcore-mprtp/03/.

[68] J. Kim, Y.-C. Chen, R. Khalili, D. Towsley, and A. Feldmann, "Multi-Source Multipath HTTP (MHTTP): A Proposal", *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 583–584, Jun. 2014, ISSN: 0163-5999. DOI: 10.1145/2637364.2592029. [Online]. Available: https://doi.org/10.1145/2637364.2592029.

[69] H. Wu, S. Ferlin, G. Caso, Ö. Alay, and A. Brunstrom, "A Survey on Multipath Transport Protocols Towards 5G Access Traffic Steering, Switching and Splitting", *IEEE Access*, vol. 9, pp. 164 417–164 439, 2021. DOI: 10.1109/ACCESS.2021.3134261.

[70] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental Evaluation of Multipath TCP Schedulers", in *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop*, ser. CSWS '14, Chicago, Illinois, USA: Association for Computing Machinery, 2014, pp. 27–32, ISBN: 9781450329910. DOI: 10.1145/2630088.2631977. [Online]. Available: https://doi.org/10.1145/2630088.2631977.

[71] T. De Schepper, J. Struye, E. Zeljković, S. Latré, and J. Famaey, "Software-defined multipath-TCP for smart mobile devices", in *2017 13th International Conference on Network and Service Management (CNSM)*, Nov. 2017, pp. 1–6. DOI: 10.23919/CNSM.2017.8256043.

[72] K. W. Choi, Y. S. Cho, Aneta, J. W. Lee, S. M. Cho, and J. Choi, "Optimal load balancing scheduler for MPTCP-based bandwidth aggregation in heterogeneous wireless environments", *Computer Communications*, vol. 112, pp. 116–130, 2017, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2017.08.018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366417302426.

[73] Y.-s. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, "ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths", in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '17, Incheon, Republic of Korea: Association for Computing Machinery, 2017, pp. 147–159, ISBN: 9781450354226. DOI: 10.1145/3143361.3143376. [Online]. Available: https://doi.org/10.1145/3143361.3143376.

[74] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli, "DAPS: Intelligent delay-aware packet scheduling for multipath transport", in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 1222–1227. DOI: 10.1109/ICC.2014.6883488.

[75] S. Ferlin, Ö. Alay, O. Mehani, and R. Boreli, "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks", in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, May 2016, pp. 431–439. DOI: 10.1109/IFIPNetworking.2016.7497206.

[76] F. Yang, Q. Wang, and P. D. Amer, "Out-of-Order Transmission for In-Order Arrival Scheduling for Multipath TCP", in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, May 2014, pp. 749–752. DOI: 10.1109/WAINA.2014.122.

[77] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, Ö. Alay, and N. Kuhn, "Low-Latency Scheduling in MPTCP", *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 302–315, Feb. 2019, ISSN: 1558-2566. DOI: 10.1109/TNET.2018.2884791.

[78] H. Wu, ". Alay, A. Brunstrom, S. Ferlin, and G. Caso, "Peekaboo: Learning-Based Multipath Scheduling for Dynamic Heterogeneous Environments", *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2295–2310, Oct. 2020, ISSN: 1558-0008. DOI: 10.1109/JSAC.2020.3000365.

[79] I. Lopez, M. Aguado, C. Pinedo, and E. Jacob, "SCADA Systems in the Railway Domain: Enhancing Reliability through Redundant MultipathTCP", in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 2305–2310. DOI: 10.1109/ITSC.2015.372.

[80] A. Frommgen, T. Erbshäußer, A. Buchmann, T. Zimmermann, and K. Wehrle, "ReMP TCP: Low latency multipath TCP", in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7. DOI: 10.1109/ICC.2016.7510787.

[81] A. Frömmgen, "Programming Models and Extensive Evaluation Support for MPTCP Scheduling, Adaptation Decisions, and DASH Video Streaming", en, Ph.D. dissertation, Technische Universität, Darmstadt, 2018. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/7709/.

[82] W. Lu, D. Yu, M. Huang, and B. Guo, "PO-MPTCP: Priorities-Oriented Data Scheduler for Multimedia Multipathing Services", *International Journal of Digital Multimedia Broadcasting*, vol. 2018, p. 1 413 026, Dec. 2018, ISSN: 1687-7578. DOI: 10.1155/2018/1413026. [Online]. Available: https://doi.org/10.1155/2018/1413026.

[83] D. Jurca and P. Frossard, "Video Packet Selection and Scheduling for Multipath Streaming", *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 629–641, Apr. 2007, ISSN: 1941-0077. DOI: 10.1109/TMM.2006.888017.

[84] D. Jurca and P. Frossard, "Distortion Optimized Multipath Video Streaming", *Proceedings of the International Packet Video Workshop*, International Journal of Imaging Systems and Technology, 2004. [Online]. Available: http://infoscience.epfl.ch/record/87129.

[85] X. Corbillon, R. Aparicio-Pardo, N. Kuhn, G. Texier, and G. Simon, "Cross-Layer Scheduler for Video Streaming over MPTCP", in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16, Klagenfurt, Austria: Association for Computing Machinery, 2016, pp. 1–12, ISBN: 9781450342971. DOI: 10.1145/2910017.2910594. [Online]. Available: https://doi.org/10.1145/2910017.2910594.

[86] Y. P. G. Chowrikoppalu, "Multipath Adaptive Video Streaming over Multipath TCP", M.S. thesis, University of Saarland, Mar. 2013. [Online]. Available: https://www.nt.uni-saarland.de/wp-content/uploads/2019/05/MasterThesisYash.pdf.

[87] J. Apostolopoulos, T. Wong, W.-t. Tan, and S. Wee, "On multiple description streaming with content delivery networks", in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, Jun. 2002, 1736–1745 vol.3. DOI: 10.1109/INFCOM.2002.1019427.

[88] S. Mao, S. Lin, S. Panwar, Y. Wang, and E. Celebi, "Video transport over ad hoc networks: multistream coding with multipath transport", *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1721–1737, Dec. 2003, ISSN: 1558-0008. DOI: 10.1109/JSAC.2003.815965.

[89] G. Xie, M. N. S. Swamy, and M. O. Ahmad, "Optimal Packet Scheduling for Multi-Description Multi-Path Video Streaming Over Wireless Networks", in *2007 IEEE International Conference on Communications*, Jun. 2007, pp. 1618–1623. DOI: 10.1109/ICC.2007.271.

[90] R. Matsufuji, D. Cavendish, K. Kumazoe, D. Nobayashi, and T. Ikenaga, "Multipath TCP path schedulers for streaming video", in *2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, Aug. 2017, pp. 1–6. DOI: 10.1109/PACRIM.2017.8121920.

[91] S. Maheshwari, P. Lundrigan, and S. K. Kasera, "Scheduling Virtual Wifi Interfaces for High Bandwidth Video Upstreaming Using Multipath TCP", in *Proceedings of the 20th International Conference on Distributed Computing and Networking*, ser. ICDCN '19, Bangalore, India: Association for Computing Machinery, 2019, pp. 1–10, ISBN: 9781450360944. DOI: 10.1145/3288599.3288620. [Online]. Available: https://doi.org/10.1145/3288599.3288620.

[92] B. Han, F. Qian, L. Ji, and V. Gopalakrishnan, "MP-DASH: Adaptive Video Streaming Over Preference-Aware Multipath", in *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '16, Irvine, California, USA: Association for Computing Machinery, 2016, pp. 129–143, ISBN: 9781450342926. DOI: 10.1145/2999572.2999606. [Online]. Available: https://doi.org/10.1145/2999572.2999606.

[93] A. Elgabli, K. Liu, and V. Aggarwal, "Optimized Preference-Aware Multi-Path Video Streaming with Scalable Video Coding", *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 159–172, Jan. 2020, ISSN: 1558-0660. DOI: 10.1109/TMC.2018.2889039.

[94] B. Y. L. Kimura, D. C. S. F. Lima, and A. A. F. Loureiro, "Packet Scheduling in Multipath TCP: Fundamentals, Lessons, and Opportunities", *IEEE Systems Journal*, vol. 15, no. 1, pp. 1445–1457, Mar. 2021, ISSN: 1937-9234. DOI: 10.1109/JSYST.2020.2965471.

[95] S. Afzal, V. Testoni, C. E. Rothenberg, P. Kolan, and I. Bouazizi, *A Holistic Survey of Wireless Multipath Video Streaming*, 2019. DOI: 10.48550/ARXIV.1906.06184. [Online]. Available: https://arxiv.org/abs/1906.06184.

[96] 3GPP, "Wireless and wireline convergence access support for the 5G System (5GS)", 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.316, Jun. 2022, Version 16.8.0.

[97] BBF, "Hybrid Access Broadband Network Architecture", Broadband Forum, Technical Report (TR) 348, Jul. 2016, Issue 1.

[98] E. Blanton, D. V. Paxson, and M. Allman, *TCP Congestion Control*, RFC 5681, Sep. 2009. DOI: 10.17487/RFC5681. [Online]. Available: https://www.rfc-editor.org/info/rfc5681.

[99] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, and R. Scheffenegger, *CUBIC for Fast Long-Distance Networks*, RFC 8312, Feb. 2018. DOI: 10.17487/RFC8312. [Online]. Available: https://www.rfc-editor.org/info/rfc8312.

[100] N. Cardwell, Y. Cheng, S. H. Yeganeh, I. Swett, and V. Jacobson, "BBR Congestion Control", Internet Engineering Task Force, Internet-Draft draft-cardwell-iccrg-bbr-congestion-control-02, Mar. 2022, Work in Progress, 66 pp. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control-02.

[101] S. Hemminger, *[TCP]: make cubic the default*, in collab. with D. Miller, version 597811ec167fa01c926a0957a91d9e39baa30e64, Sep. 24, 2006. [Online]. Available: https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/.

[102] Apple, *xnu-2782.1.97*, version a3bb9fcc43a00154884a30c9080595284c26cec9, Oct. 24, 2014. [Online]. Available: https://github.com/apple-oss-distributions/xnu/blob/main/bsd/netinet/tcp_cc.h.

[103] P. Balasubramanian, *Updates on Windows TCP*, Nov. 2017. [Online]. Available: https://datatracker.ietf.org/meeting/100/materials/slides-100-tcpm-updates-on-windows-tcp.

[104] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-Based Congestion Control", *Commun. ACM*, vol. 60, no. 2, pp. 58–66, Jan. 2017, ISSN: 0001-0782. DOI: 10.1145/3009824. [Online]. Available: https://doi.org/10.1145/3009824.

[105] N. Cardwell *et al.*, *BBR Congestion Control Work at Google IETF 101 Update*, Mar. 2018. [Online]. Available: https://datatracker.ietf.org/meeting/101/materials/slides-101-iccrg-an-update-on-bbr-work-at-google-00.

[106] M. Becke, T. Dreibholz, H. Adhari, and E. P. Rathgeb, "On the fairness of transport protocols in a multi-path environment", in *2012 IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 2666–2672. DOI: 10.1109/ICC.2012.6363695.

[107] C. Xu, J. Zhao, and G.-M. Muntean, "Congestion Control Design for Multipath Transport Protocols: A Survey", *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2948–2969, Apr. 2016, ISSN: 1553-877X. DOI: 10.1109/COMST.2016.2558818.

[108] C. Raiciu, M. J. Handley, and D. Wischik, *Coupled Congestion Control for Multipath Transport Protocols*, RFC 6356, Oct. 2011. DOI: 10.17487/RFC6356. [Online]. Available: https://www.rfc-editor.org/info/rfc6356.

[109] R. Khalili, N. Gast, M. Popovic, and J.-Y. L. Boudec, "Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP", Internet Engineering Task Force, Internet-Draft draft-khalili-mptcp-congestion-control-05, Jul. 2014, Work in Progress, 11 pp. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-khalili-mptcp-congestion-control-05.

[110] A. Walid, Q. Peng, J. Hwang, and S. H. Low, "Balanced Linked Adaptation Congestion Control Algorithm for MPTCP", Internet Engineering Task Force, Internet-Draft draft-walid-mptcp-congestion-control-04, Jan. 2016, Work in Progress, 11 pp. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-walid-mptcp-congestion-control-04.

[111] M. Xu, Y. Cao, and E. Dong, "Delay-based Congestion Control for MPTCP", Internet Engineering Task Force, Internet-Draft draft-xu-mptcp-congestion-control-05, Jan. 2017, Work in Progress, 12 pp. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-xu-mptcp-congestion-control-05.

[112] F. Fu, X. Zhou, T. Dreibholz, K. Wang, F. Zhou, and Q. Gan, "Performance comparison of congestion control strategies for multi-path TCP in the NORNET testbed", in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, Nov. 2015, pp. 1–6. DOI: 10.1109/ICCChina.2015.7448667.

[113] ITU-T, "Advanced video coding for generic audiovisual services", International Telecommunication Union, Geneva, Recommendation H.264, Aug. 2021.

[114] ITU-T, "High efficiency video coding", International Telecommunication Union, Geneva, Recommendation H.265, Aug. 2021.

[115] P. Wilkins, Y. Xu, L. Quillio, J. Bankoski, J. Salonen, and J. Koleszar, *VP8 Data Format and Decoding Guide*, RFC 6386, Nov. 2011. DOI: 10.17487/RFC6386. [Online]. Available: https://rfc-editor.org/rfc/rfc6386.txt.

[116] A. Grange, P. de Rivaz, and J. Hunt, *VP9 Bitstream & Decoding Process Specification v0.6*, The WebM Project, Online available at: https://www.webmproject.org/vp9/, accessed at 20/12/2021, Mar. 2016.

[117] "Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats", International Organization for Standardization, Geneva, CH, Standard, Dec. 2019.

[118] R. Pantos and W. May, *HTTP Live Streaming*, RFC 8216, Aug. 2017. DOI: 10.17487/RFC8216. [Online]. Available: https://rfc-editor.org/rfc/rfc8216.txt.

[119] K. Yamagishi, "QoE-estimation models for video streaming services", in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec. 2017, pp. 357–363. DOI: 10.1109/APSIPA.2017.8282058.

[120] R. R. R. Rao *et al.*, "Adaptive video streaming with current codecs and formats: Extensions to parametric video quality model itu-t p.1203", *Electronic Imaging*, vol. 31, no. 10, pp. 314-1–314-1, 2019. DOI: 10.2352/ISSN.2470-1173.2019.10.IQSP-314. [Online]. Available: https://library.imaging.org/ei/articles/31/10/art00015.

[121] S. Göring, A. Raake, and B. Feiten, "A framework for QoE analysis of encrypted video streams", in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, 2017, pp. 1–3. DOI: 10.1109/QoMEX.2017.7965640.

[122] W. Robitza *et al.*, "HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P. 1203: Open Databases and Software", in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18, Amsterdam, Netherlands: Association for Computing Machinery, 2018, pp. 466–471, ISBN: 9781450351928. DOI: 10.1145/3204949.3208124. [Online]. Available: https://doi.org/10.1145/3204949.3208124.

[123] G. Miranda, D. F. Macedo, and J. M. Marquez-Barja, "Estimating Video on Demand QoE from Network QoS through ICMP Probes", *IEEE Transactions on Network and Service Management*, pp. 1–1, 2021. DOI: 10.1109/TNSM.2021.3129610.

[124] ITU-T, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport", International Telecommunication Union, Geneva, Recommendation P.1203, Oct. 2017.

[125] ITU-T, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Video quality estimation module", International Telecommunication Union, Geneva, Recommendation P.1203.1, Jan. 2019.

[126] ITU-T, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Audio quality estimation module", International Telecommunication Union, Geneva, Recommendation P.1203.2, Oct. 2017.

[127] ITU-T, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Quality integration module", International Telecommunication Union, Geneva, Recommendation P.1203.3, Jan. 2019.

[128] N. McCormick, "The Future of Mobile Tariffs: 2030", Omdia, Tech. Rep., Jan. 2022.

[129] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology", *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 963–977, 2004. DOI: 10.1109/TNET.2004.838606.

[130] F. Zhou, T. Dreibholz, X. Zhou, F. Fu, Y. Tan, and Q. Gan, "The Performance Impact of Buffer Sizes for Multi-path TCP in Internet Setups", in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, Mar. 2017, pp. 9–16. DOI: 10.1109/AINA.2017.26.

[131] G. Detal, C. Paasch, and O. Bonaventure, "Multipath in the Middle(Box)", in *Proceedings of the 2013 Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, ser. HotMiddlebox '13, Santa Barbara, California, USA: Association for Computing Machinery, 2013, pp. 1–6, ISBN: 9781450325745. DOI: 10.1145/2535828.2535829. [Online]. Available: https://doi.org/10.1145/2535828.2535829.

[132] O. Bonaventure, M. Boucadair, S. Gundavelli, S. Seo, and B. Hesmans, *0-RTT TCP Convert Protocol*, RFC 8803, Jul. 2020. DOI: 10.17487/RFC8803. [Online]. Available: https://www.rfc-editor.org/info/rfc8803.

[133] Proximus. "Tessares-Proximus' Access Bonding - offering faster Internet in large, sparsely populated rural areas – now successfully qualified to move to a countrywide deployment phase.", Proximus. (May 17, 2017), [Online]. Available: https://web.archive.org/web/20180705233839/https://www.proximus.com/en/news/tessares-proximus%E2%80%99-access-bonding-offering-faster-internet-large-sparsely-populated-rural-areas (visited on 10/03/2023).

[134] Ian Scales. "Wire in the mud: Dutch rural users offered LTE boost for slow copper", Telecom TV. (May 10, 2019), [Online]. Available: https://www.telecomtv.com/content/broadband/wire-in-the-mud-dutch-rural-users-offered-lte-boost-for-slow-copper-35114/ (visited on 10/03/2023).

[135] telecompaper. "Telia intros hybrid fixed-mobile broadband service in Finland ", telecompaper. (Aug. 17, 2018), [Online]. Available: https://www.telecompaper.com/news/telia-intros-hybrid-fixed-mobile-broadband-service-in-finland--1257235 (visited on 10/03/2023).

[136] Mark Jackson. "BT Launch Hybrid 4G Speed Boost for SME Copper Broadband Lines", ISPreview. (May 24, 2022), [Online]. Available: https://www.ispreview.co.uk/index.php/2022/05/bt-launch-hybrid-4g-speed-boost-for-sme-copper-broadband-lines.html (visited on 10/03/2023).

[137] MediaTek. "MediaTek to Showcase 5G, Satellite Communications, Computing and Connectivity Technology Advancements at MWC", MediaTek. (Feb. 22, 2023), [Online]. Available: https://corp.mediatek.com/news-events/press-releases/mediatek-to-showcase-5g-satellite-communications-computing-and-connectivity-technology-advancements-at-mwc (visited on 02/22/2023).

Appendix A

Possible return values of the COM and CPF decision logic

Depending on the configuration, scheduler state and path state, the returned path for data transmission in the *CPF* and *COM* scheduler varies following the flow diagrams in Figure 3.2 (CPF) and in Figure 3.15 (COM).

As an example, the results of this process are shown for a system with two and three paths over all relevant combinations for *CPF* in Table A.1 and Table A.3 and for *COM* in Table A.2 and Table A.4.

For readability, the paths in the two-path system are labeled Wi-Fi and Cellular, with Cellular having a higher cost. In the three path system, a third path Satellite is added with highest cost. While the path selection of *CPF* is depending on the configured cost and path availability, *COM* differentiates the path availability into the states broken or congested as described in section 3.6. Also the path block state is considered as the essence of the *COM* algorithm. It should be noted in the tables that the cheapest path in *COM* does not hold a block status by definition, and if a path is considered broken, a congestion status is not available and a block status is useless for evaluation.

In case no path is available for data transmission, e.g., due to broken or congested links, a return value of `Null` illustrates this fact.

Two path scenario

Table A.1: Possible return values of CPF in a two path system

Path	Cost/Prio	Available	Return
Wi-Fi	0	no	Null
Cellular	1	no	
Wi-Fi	0	no	Cellular
Cellular	1	yes	
Wi-Fi	0	yes	WiFi
Cellular	1	no	
Wi-Fi	0	yes	WiFi
Cellular	1	yes	

Table A.2: Possible return values of COM in a two path system

Path	Cost/Prio	Broken	Congested	Blocked	Return
Wi-Fi	0	no	no		WiFi
Cellular	1	no	no	no	
Wi-Fi	0	no	no		WiFi
Cellular	1	no	no	yes	
Wi-Fi	0	no	yes		Cellular
Cellular	1	no	no	no	
Wi-Fi	0	no	yes		Null
Cellular	1	no	no	yes	
Wi-Fi	0	no	no		WiFi
Cellular	1	no	yes	no	
Wi-Fi	0	no	no		WiFi
Cellular	1	no	yes	yes	

Continued on next page

Table A.2: Possible return values of COM in a two path system (Continued)

Path	Cost/Prio	Broken	Congested	Blocked	Return
Wi-Fi	0	no	yes		Null
Cellular	1	no	yes	no	
Wi-Fi	0	no	yes		Null
Cellular	1	no	yes	yes	
Wi-Fi	0	yes			Cellular
Cellular	1	no	no	no	
Wi-Fi	0	yes			WiFi
Cellular	1	no	no	yes	
Wi-Fi	0	yes			Null
Cellular	1	no	yes	no	
Wi-Fi	0	yes			Null
Cellular	1	no	yes	yes	
Wi-Fi	0	no	no		WiFi
Cellular	1	yes			
Wi-Fi	0	no	yes		Null
Cellular	1	yes			
Wi-Fi	0	yes			Null
Cellular	1	yes			

Three path scenario

Table A.3: Possible return values of CPF in a three path system

Path	Cost/Prio	Available	Return
Wi-Fi	0	no	Null
Cellular	1	no	
Satellite	2	no	
Wi-Fi	0	no	Satellite
Cellular	1	no	
Satellite	2	yes	
Wi-Fi	0	no	Cellular
Cellular	1	yes	
Satellite	2	no	
Wi-Fi	0	no	Cellular
Cellular	1	yes	
Satellite	2	yes	
Wi-Fi	0	yes	WiFi
Cellular	1	no	
Satellite	2	no	
Wi-Fi	0	yes	WiFi
Cellular	1	no	
Satellite	2	yes	
Wi-Fi	0	yes	WiFi
Cellular	1	yes	
Satellite	2	no	

Continued on next page

Table A.3: Possible return values of CPF in a three path system (Continued)

Path	Cost/Prio	Available	Return
Wi-Fi	0	yes	WiFi
Cellular	1	yes	
Satellite	2	yes	

Table A.4: Possible return values of COM in a three path system

Path	Cost/Prio	Broken	Congested	Blocked	Return
Wi-Fi	0	no	no		WiFi
Cellular	1	Irrelevant as long as Wi-Fi can carry data			
Satellite	2				
Wi-Fi	0	no	yes		Cellular
Cellular	1	no	no	no	
Satellite	2	Irrelevant as long as Cellular can carry data			
Wi-Fi	0	no	yes		Satellite
Cellular	1	no	no	yes	
Satellite	2	no	no	no	
Wi-Fi	0	no	yes		Null
Cellular	1	no	no	yes	
Satellite	2	no	no	yes	
Wi-Fi	0	no	yes		Satellite
Cellular	1	no	yes	no	
Satellite	2	no	no	no	
Wi-Fi	0	no	yes		Null
Cellular	1	no	yes	no	
Satellite	2	no	no	yes	

Continued on next page

Table A.4: Possible return values of COM in a three path system (Continued)

Path	Cost/Prio	Broken	Congested	Blocked	Return
Wi-Fi	0	no	yes		Satellite
Cellular	1	no	yes	yes	
Satellite	2	no	no	no	
Wi-Fi	0	no	yes		Null
Cellular	1	no	yes	yes	
Satellite	2	no	no	yes	
Wi-Fi	0	no	yes		Null
Cellular	1	no	yes	Irrelevant	
Satellite	2	no	yes		
Wi-Fi	0	yes	Irrelevant		Cellular
Cellular	1	no	no	no	
Satellite	2	Irrelevant as long as Cellular can carry data			
Wi-Fi	0	yes	Irrelevant		Satellite
Cellular	1	no	no	yes	
Satellite	2	no	no	no	
Wi-Fi	0	yes	Irrelevant		Cellular
Cellular	1	no	no	yes	
Satellite	2	no	no	yes	
Wi-Fi	0	yes	Irrelevant		Satellite
Cellular	1	no	yes	no	
Satellite	2	no	no	no	
Wi-Fi	0	yes	Irrelevant		Null
Cellular	1	no	yes	no	
Satellite	2	no	no	yes	

Continued on next page

Table A.4: Possible return values of COM in a three path system (Continued)

Path	Cost/Prio	Broken	Congested	Blocked	Return
Wi-Fi	0	yes	Irrelevant		
Cellular	1	no	yes	yes	Satellite
Satellite	2	no	no	no	
Wi-Fi	0	yes	Irrelevant		
Cellular	1	no	yes	yes	Satellite
Satellite	2	no	no	yes	
Wi-Fi	0	yes	Irrelevant		
Cellular	1	no	yes	Irrelevant	Null
Satellite	2	no	yes		
Wi-Fi	0	yes	Irrelevant		
Cellular	1	yes	Irrelevant		Satellite
Satellite	2	no	no	no	
Wi-Fi	0	yes	Irrelevant		
Cellular	1	yes	Irrelevant		Satellite
Satellite	2	no	no	yes	
Wi-Fi	0	yes	Irrelevant		
Cellular	1	yes	Irrelevant		Null
Satellite	2	no	yes	no	
Wi-Fi	0	yes	Irrelevant		
Cellular	1	yes	Irrelevant		Null
Satellite	2	no	yes	yes	
Wi-Fi	0	yes	Irrelevant		
Cellular	1	no	no	no	Cellular
Satellite	2	yes	Irrelevant		

Continued on next page

Table A.4: Possible return values of COM in a three path system (Continued)

Path	Cost/Prio	Broken	Congested	Blocked	Return
Wi-Fi	0	yes	Irrelevant		Cellular
Cellular	1	no	no	yes	
Satellite	2	yes	Irrelevant		
Wi-Fi	0	yes	Irrelevant		Null
Cellular	1	no	yes	no	
Satellite	2	yes	Irrelevant		
Wi-Fi	0	yes	Irrelevant		Null
Cellular	1	no	yes	yes	
Satellite	2	yes	Irrelevant		
Wi-Fi	0	yes	Irrelevant		Null
Cellular	1	yes	Irrelevant		
Satellite	2	yes	Irrelevant		

Appendix B

Author's additional publications

All Multipath patent families

Table B.1: All Multipath patent families of the author

No.	Patent	EU appl.	US appl.	CN appl.
1	Packet conversion device and method for allowing transparent packet-based multipath bundling	EP3119057A1		
2	Techniques for establishing a communication connection between two network entities via different network flows	EP3276891B1	US10530644B2	
3	Techniques for efficient multipath transmission	EP3522479B1	US11159423B2	CN111699666B
4	Techniques for packet data conversion	EP3534587B1	US11165893B2	CN111788812B

Continued on next page

Table B.1: All Multipath patent families of the author (Continued)

No.	Patent	EU appl.	US appl.	CN appl.
5	Data manager for distributing data of a data stream of a user equipment via multiple wireless local area network links	EP3518578B1		
6	Data flow manager for distributing data for a data stream of a user equipment, communication system and method	EP3518576B1	US10873531B2	
7	Techniques for efficient reordering of data packets in multipath scenarios	EP3531637B1	US11159442B2	CN111801915B
8	Techniques for interaction between network protocols	EP3534586B1	US11469990B2	
9	Techniques for policy management of multi-connectivity network protocols	EP3534574B1	US11582143B2	CN111837367B
10	Techniques for scheduling multipath data traffic	EP3544332B1	US11451486B2	CN111869257A
11	Techniques for multipath bundling and determining wi-fi connections for multipath bundling	EP3544363B1	US2021127440A1	CN111869311A

Continued on next page

Table B.1: All Multipath patent families of the author (Continued)

No.	Patent	EU appl.	US appl.	CN appl.
12	A communication system for transmitting a transmission control protocol segment over a communication network using a multipath transmission control protocol, corresponding method and computer program	EP3579500B1	US11329908B2	CN112262552B
13	Data flow manager for load-balancing data for a data stream of a user equipment, communication system and method	EP3772202A1	US2022279383A1	CN114223184A
14	Techniques for detecting bursty traffic pattern detection and scheduling multipath data traffic	EP3796604B1	US2022393968A1	CN114424506A
15	Route and interface selection techniques for multi-connectivity network protocols	EP3817305A1		
16	A method and communication device for transmitting multiple data streams of different communication services over a multipath transmission system	EP4042653A1	US2022393970A1	CN114503525A

Continued on next page

Table B.1: All Multipath patent families of the author (Continued)

No.	Patent	EU appl.	US appl.	CN appl.
17	Multi-connectivity capable network device and communication systems for centrally controlling multiple access of a multi-connectivity capable customer equipment to a data network and/or to services provided by the data network	EP3866430A1		
18	Multipath capable network device and communication systems for centrally monitoring and controlling data traffic to and/or from a multipath capable customer equipment	EP3866429B1		
19	Energy saving techniques for multi-connectivity devices	EP3817319B1	US2022377673A1	CN114631352B
20	Reliability and aggregation selection for multi-connectivity network protocols	EP3817304B1	US2022400081A1	CN114616808B
21	Method and network device for multi-path communication	EP3820088A1	US2022407799A1	CN114631297A

Continued on next page

Table B.1: All Multipath patent families of the author (Continued)

No.	Patent	EU appl.	US appl.	CN appl.
22	Selectable tunnel encryption level management for multi access user equipment	EP3923611A1	US2021385648A1	
23	Method and communication system for ensuring secure communication in a zero touch connectivity-environment	EP3923612A1	US2021385656A1	
24	Method for enabling zero touch connectivity (ztc) access in a communication system	EP3902303B1		
25	Internet access provider with an independent multi-connectivity framework	EP3902209A1		
26	Access to a home network within a multi-connectivity framework	EP3920509B1	US2021385894A1	
27	Domain name request multiplication for multipath-access	EP3937470A1	US11374896B2	
28	Multi-connectivity restriction for services demanding single-path or single access	EP3930376A1	US2021400563A1	

Continued on next page

Table B.1: All Multipath patent families of the author (Continued)

No.	Patent	EU appl.	US appl.	CN appl.
29	A data traffic control device, a residential router, an operator network device and a telecommunication system	EP3968577A1	US11799802B2	
30	Multipath-capable communication device	EP3968578A1	US2022086094A1	
31	Method and system for reachability of services specific to one specific network access over a different network access and system thereof	EP4002766A1		
32	Domain name system in combination with a multipath network protocol	EP3937469A1		
33	Identification of cascaded multi-connectivity and mitigation of cascaded multi-connectivity interference effects	EP3958612B1		CN115968562A
34	Alternative cellular connectivity for mobile terminals	EP3982607B1		
35	Method of policy exchange for atsss overlay and system	EP4012977A1		
36	System and method for multipath transmission	EP4080836B1		

Continued on next page

Table B.1: All Multipath patent families of the author (Continued)

No.	Patent	EU appl.	US appl.	CN appl.
37	System and method for multipath transmission with efficient adjustable reliability	EP4075742A1		
38	Techniques for an efficient use of redundant multipath scheduling	EP4044526A1		
39	Techniques for providing a generic multipath system by a flexible selection of network protocols	EP4142264A1		
40	Method for traffic matching in terminals with ue route selection policy (ursp)	EP4102787A1		
41	Determining optimized data rates for a communication path	EP4221144A1		
42	Mp-Dccp Proxy to Enable Multipath Transmission of Dccp Data Packets Between a Sender and a Receiver	EP4246937A1		
43	Techniques to Decrease the Level of Encryption in a Trusted Communication Environment	EP4246883A1		
44	Multipath Receiver and Processing of 3GPP ATSSS on a multipath receiver	EP4254900A1		