Adaptive Multipath Scheduling for 5G Networks and Beyond: A Learning Perspective

Hongjia Wu

OsloMet Avhandling 2022 nr 6

OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITETET

Adaptive Multipath Scheduling for 5G Networks and Beyond: A Learning Perspective

Hongjia Wu

Dissertation for the degree of Philosophiae Doctor (PhD)

Simula Metropolitan Center for Digital Engineering
Faculty of Technology, Art and Design
OsloMet – Oslo Metropolitan University

Spring 2022

OsloMet Avhandling 2022 nr 6

ISSN 2535-471X (trykt)
ISSN 2535-5414 (online)

ISBN 978-82-8364-377-0 (trykt)
ISBN 978-82-8364-408-1 (online)

OsloMet – storbyuniversitetet
Universitetsbiblioteket
Skriftserien
St. Olavs plass 4,
0130 Oslo,
Telefon (47) 64 84 90 00

Postadresse:
Postboks 4, St. Olavs plass
0130 Oslo

Trykket hos Byråservice

Trykket på Scandia 2000 white, 80 gram på materiesider/200 gram på coveret

Acknowledgements

This thesis is submitted for the degree of Philosophiae Doctor. This work is to the best of my knowledge original, except where references are made to the previous work.

The following document uses the first-person plural to indicate the collaborative nature of much of this work. In particular, Özgü Alay and Anna Brunstrom have been constant sources of inspiration and wisdom. They have provided invaluable advice on many topics and their formidable expertise has time and again proven to be priceless. I could not have imagined having better advisors for this work. I would also like to thank Giuseppe Caso and Simone Ferlin for their keen insights and tons of constructive suggestions.

The results presented in this work utilize the tool of *Mininet*. I would like to thank the developers of *Mininet* and the members of *Mininet* Mailing list for their feedback and ideas. The algorithms presented in this work are set up upon the library of *mpquic-go*. I would like to thank Quentin De Coninck and Olivier Bonaventure for developing the idea of multipath QUIC into the practical technique.

I would like to thank Mengdi Liu for her enormous depths of support that have gone well beyond expectations. I would like to thank Gengchen Wu for bringing joy to my life. I would also like to thank all my friends in the gym and boxing gym. Writing this thesis has never been a burden, when I know I can train like an athlete every morning.

Abstract

With the ready availability of multiple radio interfaces in today's smart devices, there is a growing number of hosts that can support data communication over two or more interfaces. Nonetheless, the classical transport protocols such as TCP, UDP, and the emerging QUIC still only utilize one interface at a time for data communication. In the past few years, the advents of multipath transport protocols begin to fill such a gap thanks to the proposals, standardizations, and deployments from academia and industry. Multipath transport protocols allow the concurrent use of multiple network paths for fast and reliable data exchange, potentially improving the performance and resilience of Internet traffic flows. Among the functionalities of multipath transport protocols, the multipath scheduler plays a key role since it controls the distribution of data packets over different network paths. Scheduling problem becomes quite challenging for existing multipath schedulers that are designed based on predefined rules considering the dynamic path characteristics (e.g., time-varying bandwidth, delay, and packet loss) of 5G and beyond networks. In this thesis, we focus on adaptive multipath scheduling algorithms, tackling the challenges brought from dynamic 5G networks and beyond.

Firstly, to understand the context of our work, we conduct a survey on multipath transport for 5G networks, where the network path can present in the form of 4G, 5G, and WLAN. Specifically, we discuss how the literature on multipath transport maps to specific 5G steering functionalities and slice service types, paving the way for applying multipath transport for 5G and beyond. The survey acknowledges the necessity to design an adaptive multipath scheduler for dynamic network paths and points out the learning-based design as a potential solution.

Next, following a learning-based design, we start from the case where dynamic network paths are heterogeneous, e.g., 4G and WLAN. To this end, we propose Peekaboo, a novel online learning-based multipath scheduler that is able to learn scheduling decisions to adopt from both deterministic and stochastic aspects. From the emulations and real-world examinations, we demonstrate the superiority of Peekaboo over the multipath schedulers based on predefined rules. However, we acknowledge, from the experiments, the superiority of Peekaboo mainly exists within the heterogeneous networks.

Then, we extend the adaptive multipath scheduler's applicable scenarios to also generic dynamic network paths, aiming for different types of path combinations that can appear in 5G.

To this end, we propose M-Peekaboo with a generic path selection scheme by generalizing the learning framework of Peekaboo. From the 5G-trace-driven emulations across both static and mobile scenarios, we demonstrate the benefits of M-Peekaboo over Peekaboo and the multipath schedulers based on predefined rules. However, due to the online learning essences of Peekaboo and M-Peekaboo, we acknowledge that neither of them can adapt sufficiently fast to match the rapidly changing networks.

Finally, to fill such a gap, we propose FALCON, which uses offline learning for deriving meta-models that are fine-tuned by online learning to improve the quality and speed of adaptation simultaneously. From the 5G-trace-driven emulations across both static and mobile scenarios and real-world examinations, FALCON is shown to have a much shorter adaptation time and better adaptation accuracy than M-Peekaboo and other newly proposed learning-based multipath schedulers that appeared during the writing of this thesis.

Sammendrag

Med den tilgjengelige tilgjengeligheten av flere radiogrensesnitt i dagens smarte enheter, er det et økende antall verter som kan støtte datakommunikasjon over to eller flere grensesnitt. Ikke desto mindre bruker de klassiske transportprotokollene som TCP, UDP og den nye QUIC fortsatt bare ett grensesnitt om gangen for datakommunikasjon. I løpet av de siste årene har fremskrittene med flerveis transportprotokoller begynt å fylle et slikt hull takket være forslagene, standardiseringene og distribusjonene fra akademia og industri. Multipath transportprotokoller tillater samtidig bruk av flere nettverksbaner for rask og pålitelig datautveksling, noe som potensielt forbedrer ytelsen og motstandskraften til internettrafikkstrømmer. Blant funksjonalitetene til flerbanetransportprotokoller spiller flerbaneplanleggeren en nøkkelrolle siden den styrer fordelingen av datapakker over forskjellige nettverksbaner. Planleggingsproblem blir ganske utfordrende for eksisterende flerbaneplanleggere som er designet basert på forhåndsdefinerte regler med tanke på de dynamiske banegenskapene (f.eks. Tidsvarierende båndbredde, forsinkelse og tap av pakker) for 5G og utover nettverk. I denne oppgaven fokuserer vi på adaptive flerbaneplanleggingsalgoritmer, og takler utfordringene fra dynamiske 5G -nettverk og videre.

For det første, for å forstå konteksten i arbeidet vårt, gjennomfører vi en undersøkelse om flerveis transport for 5G -nettverk, der nettverksbanen kan presenteres i form av 4G, 5G og WLAN. Spesielt diskuterer vi hvordan litteraturen om flerveis transportkart til spesifikke 5G -styringsfunksjoner og skive tjenestetyper, baner vei for å anvende flerveis transport for 5G og utover. Undersøkelsen erkjenner nødvendigheten av å designe en adaptiv flerbaneplanlegger for dynamiske nettverksbaner og peker på det læringsbaserte designet som en potensiell løsning.

Etter et læringsbasert design starter vi ut fra det tilfellet der dynamiske nettverksbaner er heterogene, f.eks. 4G og WLAN. For dette formål foreslår vi Peekaboo, en ny online læringsbasert flerbaneplanlegger som er i stand til å lære planleggingsbeslutninger å ta fra både deterministiske og stokastiske aspekter. Fra emuleringene og virkelige undersøkelser demonstrerer vi Peekaboo sin overlegenhet i forhold til flerbaneplanleggerne basert på forhåndsdefinerte regler. Imidlertid erkjenner vi, fra eksperimentene, at Peekaboo -overlegenhet hovedsakelig eksisterer i de heterogene nettverkene.

Deretter utvider vi de adaptive flerbaneplanleggerens gjeldende scenarier til også generiske dynamiske nettverksbaner, med sikte på forskjellige typer banekombinasjoner som kan vises i 5G. For dette formål foreslår vi M-Peekaboo med en generisk stievalgordning ved å generalisere læringsrammen til Peekaboo. Fra de 5G-spor-drevne emuleringene på tvers av både statiske og mobile scenarier demonstrerer vi fordelene med M-Peekaboo fremfor Peekaboo og flerbaneplanleggerne basert på forhåndsdefinerte regler. På grunn av de elektroniske læringsessensene til Peekaboo og M-Peekaboo, erkjenner vi imidlertid at ingen av dem kan tilpasse seg tilstrekkelig raskt for å matche de raskt skiftende nettverkene.

Til slutt, for å fylle et slikt gap, foreslår vi FALCON, som bruker offline læring for å utlede metamodeller som er finjustert av online læring for å forbedre kvaliteten og hastigheten på tilpasning samtidig. Fra 5G-spor-drevne emuleringer på tvers av både statiske og mobile scenarier og virkelige undersøkelser, har FALCON vist seg å ha en mye kortere tilpasningstid og bedre tilpasningsnøyaktighet enn M-Peekaboo og andre nylig foreslåtte læringsbaserte flerbaneplanleggere som dukket opp under skrivingen av denne oppgaven.

Table of Contents

List of Articles

Article I: Wu, Hongjia, Simone Ferlin, Giuseppe Caso, Özgü Alay, and Anna Brunstrom. "A Survey on Multipath Transport Protocols Towards 5G Access Traffic Steering, Switching and Splitting." *IEEE Access 9 (2021): 164417-164439.* DOI: **10.1109/ACCESS.2021.3134261**

Article II: Wu, Hongjia, Özgü Alay, Anna Brunstrom, Simone Ferlin, and Giuseppe Caso. "Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments." *IEEE Journal on Selected Areas in Communications 38, no. 10 (2020): 2295-2310.* DOI: **10.1109/JSAC.2020.3000365**

Article III: Wu, Hongjia, Giuseppe Caso, Simone Ferlin, Özgü Alay, and Anna Brunstrom. "Multipath Scheduling for 5G Networks: Evaluation and Outlook." *IEEE Communications Magazine 59, no. 4 (2021): 44-50.* DOI: **10.1109/MCOM.001.2000881**

Article IV: Wu, Hongjia, Özgü Alay, Anna Brunstrom, Giuseppe Caso, and Simone Ferlin. "FALCON: Fast and Accurate Multipath Scheduling using Offline and Online Learning." Submitted to *IEEE/ACM Transactions on Networking (2022)*
DOI: **https://doi.org/10.48550/arXiv.2201.08969**

1 Introduction

The Internet was originally designed as a “two-connected net”, thus guaranteeing that no single failure would cause any non-failed portion of the network to lose connectivity [1]. Practically, any source-destination pair should follow the multipath transmission paradigm to at least ensure the resiliency of the network and possibly enhance the throughput and decrease the delay, etc.

Over the years, many technologies emerge from different layers of the network stack to achieve multipath transmission. From the perspective of the transport layer, the classical transport protocol such as Transmission Control Protocol (TCP) and the emerging QUIC, however, still assume that the host can only use one interface at a time. This is in sharp contrast with the proliferation of smart devices equipped with multiple radio interfaces, e.g., the WLAN and the cellular interface. The advent of multipath transport protocols, such as Multipath TCP (MPTCP) and Multipath QUIC (MPQUIC) fills such a mismatch. The multipath transport protocols can concurrently use multiple network paths, enabling the potential of fast and reliable data communication.

Meanwhile, the 5th Generation of mobile communications (5G) raises the expectations towards connecting the whole society and exploits multiple technologies to be able to accommodate the requirements of a wide range of services. 5G centralizes three major performance aspects: very high data rates, ultra-reliable and low latency, and massive connectivity; and, correspondingly, provides three services, including enhanced Mobile Broadband (eMBB), Ultra Reliable Low-Latency Communications (URLLC), and massive Machine Type Communications (mMTC). To fulfill these performance aspects, several enhancements have been proposed both in radio access and core networks [2] [3] [4]. Multi-connectivity, as one of the prominent technologies among these, can use multiple radio access technologies at the same time. Already investigated in the pre-5G era, multi-connectivity focuses more on the solutions at the radio level. Recently, the proposal of Access Traffic Steering, Switching, and Splitting (ATSSS) at Release 16 or the 3rd Generation Partnership Project (3GPP) foresees the 5G Core network’s support for multipath transport protocols on 3GPP and non-3GPP networks [5]. Therefore, the role of multipath transport protocols in 5G is expected to become particularly significant.

1.1 Scope

Among several functionalities within the multipath transport protocol, the multipath scheduler plays the key role of efficiently distributing data over different paths. The data packets from the application reside in the send buffer, and the scheduler assigns each packet to a different interface based on a particular scheduling policy, ultimately impacting the achievable performance in terms of experienced throughput, latency, and connection reliability. In this thesis, we focus on the design of multipath scheduler for 5G networks and beyond.

1.2 Objective

The thesis aims to design a high-performing multipath scheduler that ultimately aids 5G to achieve qualified services for eMBB, URLLC, and mMTC. The design of a high-performing multipath scheduler is however a challenging problem, especially under 5G networks where, e.g., the use of millimeter wave (mmWave) [6] shapes a highly time-varying channel. Under such a condition, the multipath scheduler should adapt its scheduling policy with the time-varying channel accordingly.

Existing multipath schedulers employ predefined rules to select the best path to use under some specific network conditions and, thus, often result in a coarse-grained scheduling policy that might not fully suit the current network conditions, particularly when network conditions vary rapidly along the time scale. Under this context, the thesis specifically aims to design an adaptive multipath scheduler for 5G networks and beyond. We choose to design the adaptative multipath scheduler from the learning perspective due to its potential to learn and fit well to the given network conditions. Further, the thesis aims to validate the designed scheduler by operating upon diverse emulated and real 5G network scenarios with different applications.

1.3 Structure

The rest of the thesis is organized as follows. Section
2 presents the background of this thesis including 5G, multi-connectivity in 5G, and multipath transport protocols with different components. Section 3 outlines the research questions of this thesis and Section 4 discusses the research methodologies that we applied. Section 5 presents our contributions in the context of related work. Section 6 concludes our work, presents the limitations of our research, and, based on the limitations, presents the possible future work.

2 Background

This section presents the background of this thesis. Section 2.1 presents the bird's-eye view of 5G. Section 2.2 presents the multi-connectivity which is one of the paradigms that seeks to meet the requirements in 5G. Among different technologies utilized in multi-connectivity, the multipath transport protocol is one of the prominent technologies. Section 2.3 goes on to describe multipath transport protocols with their various components including path management, congestion control, scheduling, which is the research focus of this thesis, and reliable transfer.

2.1 Bird’s-Eye View of 5G

Early in 2012, the International Telecommunication Union Radiocommunication Sector (ITU-R) launched a program to build "International Mobile Telecommunications (IMT) for 2020 and Beyond ", laying the groundwork for 5G research and development to take off globally. In 2015, ITU-R issued the overall objective of IMT for 2020 and Beyond [7] that should be resolved in 5G. Therein, the ITU divides 5G into three services: enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine Type Communications (mMTC). From the use cases’ perspective, eMBB focuses on services with high bandwidth requirements, such as high definition (HD) video streaming and virtual/augmented reality (VR/AR) applications, to address people's demand for a more digitally connected lifestyle; URLLC focuses on latency-sensitive and high-reliability services such as assisted and automated driving, remote robotics, and mission-critical applications, to meet digital industry demands; mMTC focuses on services that require a high level of connectivity, such as smart cities and smart agriculture, to address the needs of a fully connected digital society. Figure 1 shows some hypothetical use cases for these three categories, with the topological relationship shown as a triangle.

From the key capability’s perspective, the user-experienced data rate, area traffic capacity, peak data rate, mobility, energy efficiency, and spectrum efficiency are all important factors in eMBB. Low latency is a top priority in various industrially vital applications at URLLC. Some high mobility use cases, such as transportation safety, would also necessitate this crucial capability. High connection density is required in mMTC to enable a large number of devices,

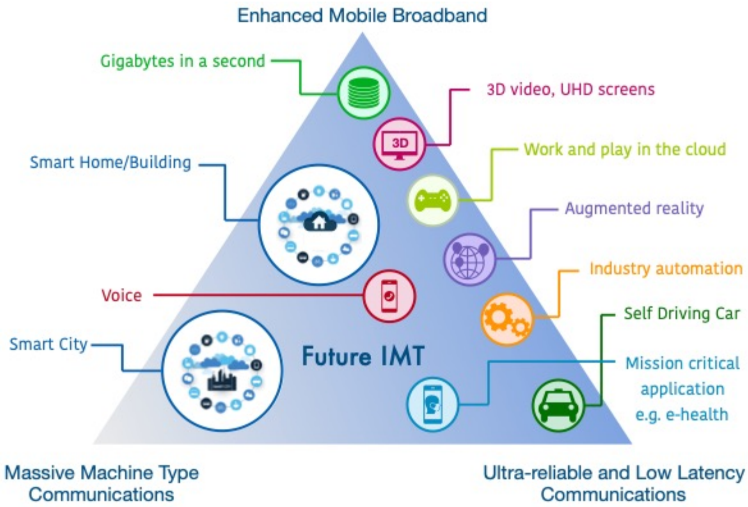


Figure 1. 5G services and corresponding reference use cases [7].

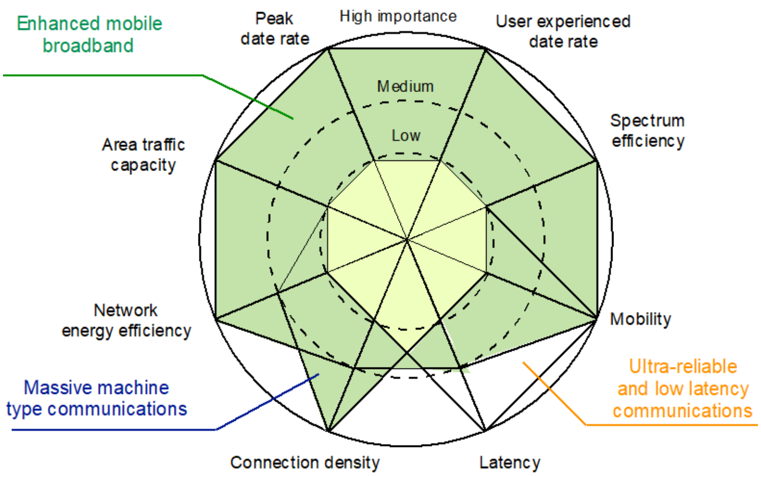


Figure 2. Key capability comparison among 5G services [7].

such as Internet of Things (IoT), that may use the radio access network intermittently to send small to large amounts of data under low mobility. Figure 2 shows the comparison of each key capability for eMBB, URLLC and mMTC.

While ITU-R sets up the general requirements of 5G, 3GPP makes the formal specifications to meet these requirements. Particularly, 3GPP adopts the concept of network slicing to map the 5G services into the Public Land Mobile Network (PLMN) with different Slice Service Types (SSTs) [5]. Different SSTs operate within the network slicing architecture that enables the multiplexing of SST-dedicated logical networks on the same physical network infrastructure [8]. Several multiplexing approaches [9] [10] [11] including Puncturing, Superposition, and Orthogonal scheduling have been proposed to achieve the objective of 5G services' coexistence.

In aforementioned 5G services, key enhancing capabilities related to throughput, latency or reliability may partially depend on the 5G system, e.g., radio frequency bands to achieve higher throughput, the radio protocol stack itself, e.g., guaranteeing that all services can coexist. Other aspects however may be tackled by improving the interconnection and intersection between the 5G system and other infrastructure services, e.g., allowing service hosting on the 5G system from services outside the Internet, or allowing 5G systems to leverage existing distributed cloud infrastructures for their own operation. From a different perspective, how UEs connect and use the 5G system can be further leveraged. One of the technologies to accomplish the latter aspect is the multi-connectivity which will be elaborated in the next subsection.

2.2 Multi-connectivity in 5G

Multi-connectivity uses multiple Radio Access Technologies (RATs) at the same time [3] to provide not only improved QoS for users, but also better load balancing between available RATs on the network side, thus helping to meet the 5G objectives.

One of the earliest multi-connectivity solutions is the Access Network Discovery and Selection Function (ANDSF), which was introduced in 3GPP Release 9 [12]. ANDSF is an optional component of the core network that provides end users with context information about non-3GPP systems (e.g., Wireless Local Area Network (WLAN)) to promote interoperability between diverse access networks. Coordinated Multi-Point (CoMP), which was introduced in 3GPP Release 11 [13], allows many base stations to transmit (or receive) the same data to a UE in parallel, increasing communication quality in locations with poor coverage. CoMP lies across physical and MAC layers. After that, Dual Connectivity (DC), introduced in 3GPP Release 12 [14], is operated in the above Packet Data Convergence Protocol (PDCP) layer and allows a UE to use two not co-located Long-Term Evolution (LTE) access nodes, e.g., two evolved Node Bs (eNBs). The Master eNB ends the control plane in the LTE core and coordinates with the Secondary eNB to give the UE additional radio resources. Similar solutions for non-3GPP access are introduced in 3GPP Release 13 [15] and are expanded in 3GPP Release 14 [16]. They are referred to as LTE-WLAN Aggregation (LWA) and LTE-WLAN radio-level integration with IP security tunnel (LWIP). The WLAN access point in both circumstances has a similar scope to a Secondary eNB in DC and can be co-located with the primary access node or not. The cellular network receives WLAN-related measurements from the user device and

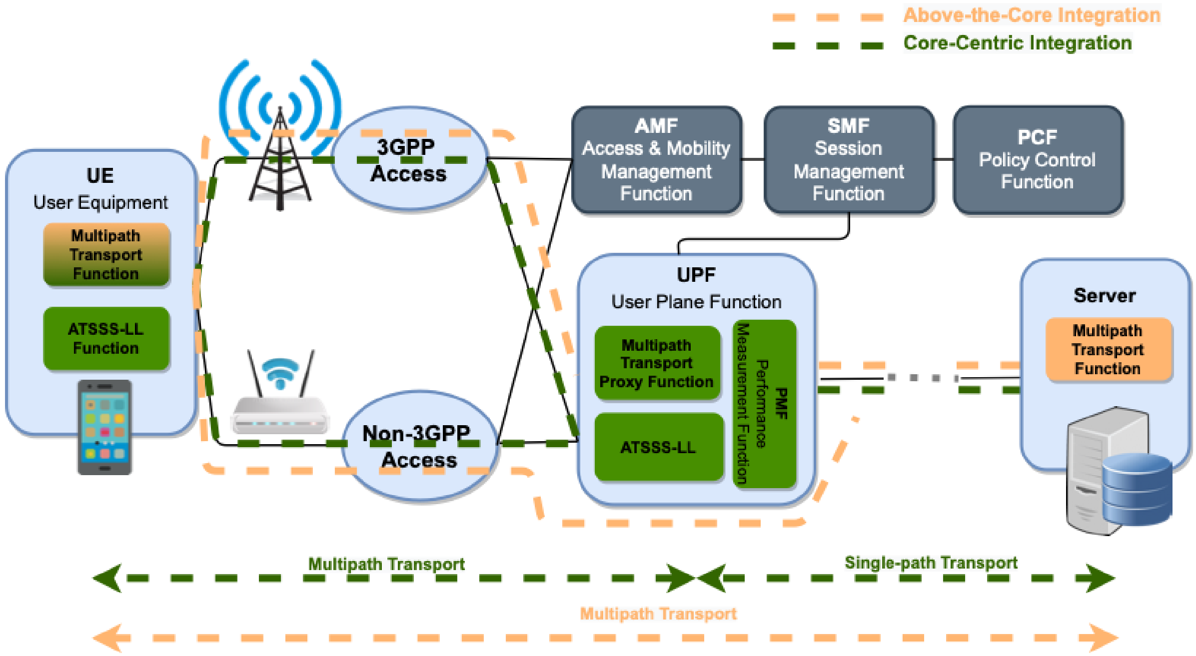


Figure 3. High-level view of Above-the-core and Core-centric integration options in 5G. For the second option, the main 5G functional blocks involved in the ATSSS architecture are reported in [5].

decides whether or not to activate the multi-connectivity option. The WLAN traffic is managed within the LTE system via specific adaptation protocols [17]. Mechanisms similar to LWA and LWIP can be envisioned for 5G [18]. However, initial proposals in 3GPP Release 15 [19] focuses on cellular access and have led to extending DC to support parallel use of LTE and 5G New Radio (NR).

Along with the above solutions at the radio level, the multipath transport protocol [20] [21] [22] can also be utilized to achieve the 5G multi-connectivity. Via multipath transport protocols, two main integration approaches are available, namely *Above-the-Core* and *Core-Centric*. In the Above-the-Core integration, the multipath transport protocol is deployed at the client and the server sides, and the aggregation of different paths occurs in between. In the Core-Centric integration, the multipath transport protocol is deployed at the client and 5G Core, and single path transport is between the core network and the server. A high-level view of both approaches is shown in Figure 3, with orange and green dashed lines representing Above-the-Core and Core-Centric, respectively. The Above-the-Core integration has been prevalent in academia and industry [23] [24] [25] [26] [27] [28] [29] [30] [31] [32], subjecting to the original end-to-end deployment architecture of multipath transport protocols. The Core-Centric integration, as highlighted by several use cases [33] [34], is a stronger candidate to be adopted in 5G systems, since it enables a more direct control of multi-connectivity within the cellular system. 3GPP has specified the ATSSS in TS 23.501 [5], as an instantiation of the Core-Centric approach. The significant concept being introduced is the Multi-Access Protocol Data Unit (MA PDU) session. The MA PDU session generalizes the single-access PDU session and allows an application to send/receive traffic over 3GPP access, non-3GPP access, or both concurrently. The MA PDU session is enabled in the ATSSS architecture. It is established between the User Equipment (UE) and User Plane Function (UPF), with both 3GPP and non-3GPP access networks in the middle. Other 5G core network functions are involved in the ATSSS operation, i.e., Access and Mobility Management Function (AMF), Session Management Function (SMF), and Policy Control function (PCF). As shown in Figure 3, the PCF controls ATSSS by delivering the policy rule to the SMF. The policy rule, shared by the SMF with the UE (uplink) or the UPF (downlink), contains the indication on which ATSSS *steering function* and *steering mode* to adopt.

With the notion of MA PDU introduced by ATSSS, there are several options for fine grained control of data flows to be served over one or more access networks. Steering selects, across several available access networks, the one that better fulfills a certain mode, e.g., smallest delay, etc. Switching takes a hard decision to abandon one of the access networks and invariably use either one access network or another, e.g., enabling connection migration and handover mechanisms. Splitting allows for using (two or more) access networks simultaneously, transferring different parts of a data flow on each available access network.

TS 23.501 defines two ways of implementing steering functionalities: a) the use of a multipath transport protocol, above the IP layer, and b) the use of a so-called ATSSS Lower Layer (ATSSS-LL), below the IP layer. In the case of multipath transport, as shown in Figure 3, the UE and UPF communicate through the Multipath Transport Function (in the UE) and the Multipath Transport Proxy Function (in the UPF). In the case of ATSSS-LL, the UE and UPF communicate with each other via the combination of ATSSS-LL Function of the UE and UPF. In addition, UPF supports Performance Measurement Functionality (PMF), that may be used by the MA PDU session to obtain access performance measurements over 3GPP and/or non-3GPP access networks.

In terms of steering modes, TS 23.501 defines four different modes that can be used with ATSSS, as follows:

- *Active-Standby*: The traffic of an MA-PDU session is sent to one access network only, referred to as "active" access. The other access network is in "standby" and takes traffic only when the active one is unavailable. The active access is defined when the MA-PDU session is established and can remain the same or change during the session lifetime;
- *Priority-based*: Some priority weights are assigned to the available access networks either statically during the establishment of a MA-PDU session or dynamically during the lifetime of a MA-PDU session. The traffic is managed by the higher priority access; however, when it is congested or unavailable, the traffic is redirected onto the lower priority access;
- *Smallest Delay*: The used access network is the one providing the shortest Round Trip Time (RTT). It conceptually belongs to the Priority-based mode but, in this case, the higher priority access is determined dynamically in the lifetime of an MA-PDU session, based on RTT measurements;
- *Load-balancing*: Each access network receives a percentage of the data of the MA-PDU session, depending on the assigned weight factor.

To sum up, the steering functionality can play the role of multipath scheduling. And the existing steering functionality is foreseen to be enriched as the existing steering functionalities only contain the basic type of scheduling approaches compared with approaches presented in the literature. The design of the multipath scheduling approach in this thesis is therefore also possible to be integrated within the 5G ATSSS as a type of steering functionality.

2.3 Multipath Transport Protocol

Multipath transport protocols leverage several network paths simultaneously and seamlessly to improve both communication throughput and resilience. Figure 4 depicts a high-level representation of the single path (left-hand side) and the multipath (right-hand side) protocol stacks. Nowadays, three main multipath protocols have been widely explored, i.e., Concurrent Multipath Transfer SCTP (CMT-SCTP) [35] [36], Multipath TCP (MPTCP) [37], and Multipath QUIC (MPQUIC) [38] [39]. Recently, Internet Engineering Task Force (IETF) has also established the working group on extending the Datagram Congestion Control Protocol (DCCP) to support the multipath operation, aiming to deliver Multipath DCCP (MP-DCCP) [40].

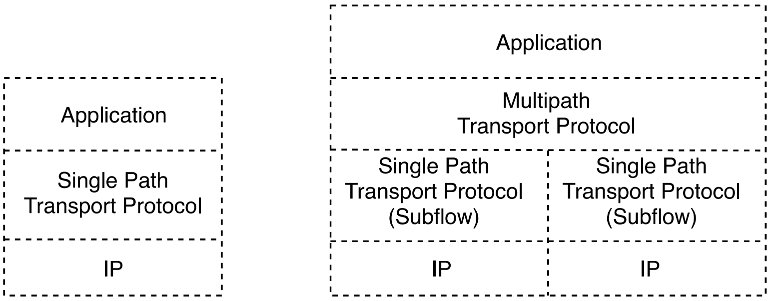


Figure 4. Single path and multipath transport protocol stack representations.

As an extension of SCTP, CMT-SCTP is one of the first multipath transport protocols. SCTP already supported some multipath capabilities, i.e., failover and mobility, but simultaneous data transfer over different paths as in CMT-SCTP was not available. Hence, CMT-SCTP adds concurrent transfer over each available path. MPTCP implements the multipath extension of the most widely used transport layer protocol, TCP. It is designed to be transparent to both higher and lower layers, in order to counteract the proliferation of middleboxes in the Internet that hinder the deployment of new transport protocols [41]. Motivated by the success of MPTCP and the interest in QUIC by both industry and academia [42] [43], MPQUIC is proposed with lots of similarities to MPTCP.

Despite different transport protocol designs and implementations, all the above mentioned multipath transport protocols share four common functionalities:

- The multipath path management, which is in charge of initiating, tearing down, and managing the connections.
- The multipath scheduling, which is in charge of distributing packets over different paths following a certain policy.
- The multipath congestion control, which aims to detect network congestion, adjust the sender rate accordingly, and deal with other aspects of the multipath transmission, e.g., fairness towards single path traffic.
- The reliable transfer, which is in charge of loss detection and loss recovery.

Path Management

The path manager component determines what path to use for connection establishment and when and how additional subflows are established, and it can also control the advertisement or acceptance of available IP addresses for new subflows. This logic generally depends on the application requirements, e.g., some applications use multipath solely for handover while others use it for load sharing. In general, however, the combination of how and when subflows are established with how the subflows are used during the connection, e.g., how packets are distributed over them, is performed in conjunction with the multipath scheduler. For instance, the path management algorithm can establish a subflow over each of two paths, and the scheduler, e.g., by means of measuring the RTT of the subflows, can prefer the subflow with the lowest RTT. This operation mode describes very closely the default path management and scheduling operations in MPTCP.

We provide an example to better understand how a path manager operates in MPTCP. Let us assume we have two hosts, i.e., Host A and B. Host A signals to Host B the support for MPTCP via a *MP_CAPABLE* TCP option during the initial handshake. Once the initial subflow is established, the *MP_JOIN* option is sent to associate a new subflow to the existing MPTCP connection. If Host A gets a new IP address during the connection, *MP_ADD* is signalled by MPTCP, telling Host B about the new address, where a new subflow can be established. For example, if Host A and Host B have initially two IP addresses each, and all possible subflows are established, the multipath connection results in a full-mesh of subflows, i.e., A1-B1, A1-B2, A2-B1, A2-B2. If Host A gets a new address, denoted A3, during the connection, it can signal this address to Host B, and additional subflows can be added to the multipath connection, i.e., A3-B1 and A3-B2.

As soon as several paths are active, a multipath sender needs to select over which path each packet will be transmitted. This selection is performed by the packet scheduler.

Scheduling

The multipath scheduler component is primarily in charge of distributing data from the application over available paths according to the given policy. Within the multipath transport protocol, the multipath scheduler works on a higher level than multipath congestion control. When the path is congested, even if the scheduler decides to send the data on that path, the data will be temporarily blocked until the Congestion Window (CWND) opens up the space. Therefore, the design of a multipath scheduler often incorporates the CWND.

The paths can be classified as homogeneous or heterogeneous, depending on how similar they are in terms of bandwidth, delay, loss rates, and other characteristics [44]. To illustrate the challenges involved in scheduling, let us consider a basic Round-Robin (RR) scheduler. In MPTCP, RR cyclically sends packets over each path, as long as there is space in CWND. While this is a very straightforward approach that may work reasonably for homogeneous paths, RR is not very useful in practice as it does not account for path heterogeneity. Since RR does not utilize any characteristics of the paths in the scheduling decision, the packets may arrive out-of-order, decreasing overall performance [45] [46]. More specifically, on one hand, the receive buffer can be flooded by out-of-order packets and the data sender can be throttled, which causes the overall throughput of the connection to be reduced or even become zero at times. This phenomenon is called receive buffer blocking. On the other hand, even though the size of the receiver buffer is not the bottleneck, the phenomenon that the application layer cannot extract the data from the receiver buffer on time due to the prior packets not having arrived yet is called head-of-line blocking.

There are different ways to tackle multipath scheduling performance challenges. For example, the scheduler can use transport layer information, e.g., RTT and CWND, to estimate the transfer time of each packet on each path. Based on the estimation, the scheduler tries to distribute packets so that they arrive in order [47] [48] [49]. Alternatively, the scheduler can duplicate packets to provide low latency or high reliability. The need depends on the current path status and the optimization goal (throughput or latency). However, the estimation accuracy of all these scheduling approaches is significantly challenged by the dynamicity presented in the networks. We elaborate further this issue and the possible solutions in Section 5.

Congestion Control

Traditional TCP congestion control algorithms operate on packet-level characteristics such as loss and delay to detect network congestion and react accordingly, e.g., by adjusting the sending rate. Among other requirements, there is a fairness notion that guarantees the same resources for each TCP flow, e.g., the same bandwidth at the shared bottleneck [50].

However, the emergence of multipath transport protocols brought the need to revisit the fairness aspect. In the case of CMT-SCTP, the protocol treats all paths belonging to a multipath connection separately, applying single-path congestion control over each path independently. In MPTCP, the fairness aspect is part of its three design goals, as discussed in [50] [51] [52] and reported as following.

- Improve Throughput: A multipath flow should perform at least as well as a single path flow would on the best available path;

- Do Not Harm: On each path, a multipath flow should not take more resources than other single path flows;
- Balance Congestion: A multipath flow should move as much traffic as possible off its most congested paths, subject to meeting the first two goals.

When it comes to MPQUIC, it is still unclear which direction standardisation will take. Initial research-oriented proposals [38] [53] suggest a design similar to MPTCP.

More generally, multipath congestion control is categorised into uncoupled and coupled approaches. The uncoupled proposals treat each of the subflows of a single multipath connection as individual connections, i.e., their CWND is increased or reduced without considering other subflows. However, for the sake of standardisation [51], the coupled proposals were adopted, since it treats all subflows belonging to the multipath connection as a single connection, subject to the concept of fairness. In MPTCP, the increase of all CWNDs of the subflows from the same multipath connection should not exceed that of a single TCP connection, thus not unfairly interacting with single path traffic. The CWND decrease, however, is handled individually, since if one of the paths is more congested than others, the subflow of the multipath connection should back-off as single-path traffic would do.

Reliable Transfer

The reliable transport layer protocols normally implement the loss detection and recovery mechanisms with the assumptions that the underlying networks may perform suboptimally in some cases, e.g., resulting in high delay and loss rates [54]. The same principle also holds in the multipath variants of these protocols. To further enhance the reliability, there is interest to apply approaches such as Forward Error Correction (FEC) and Network Coding (NC) [55] in transport protocols. In FEC, input data is encoded at the sender resulting in a combination of source and repair packets, where repair packets are used to recover lost packets at the receiver. On the other hand, NC can be performed at the sender and on intermediate nodes (all or a subset of them). In the past, different FEC and NC algorithms have been proposed inside the transport layer, in particular for TCP, where the implementations were often in conflict with the congestion control operation and prohibitively complex [56]. For multipath, FEC and NC mechanisms can be applied in the subflow level [57] [58], i.e., in the single path transport protocol connection (subflow) to alleviate the heterogeneity of the underlying paths, especially when these have different loss rates. Packets that are lost in one of the subflows can be recovered on the subflow level without retransmission. Further, since less redundancy information is required, incorporating the FEC and NC with the multipath scheduling is normally of higher efficiency than the possible packet duplication mechanism within the multipath scheduling to guarantee the reliable transfer.

3 Main Challenges

The central objective of this thesis is to design a high-performing adaptive multipath scheduler for 5G networks that can support the 5G services and use cases, as presented in Section 1. To meet this research objective, in this section, three underlying research questions of this thesis are outlined.

- Question 1: *What is the status of existing multipath schedulers within the context of 5G networks?*

  We ask this question to motivate the necessity to design an adaptative multipath scheduler. This question is discussed in Paper I. To have a better understanding of this question, we review the state of the art in the whole area of multipath transport protocols and map it into the context of 5G.

- Question 2: *How to design an adaptative multipath scheduler?*

  We ask this question to set up the foundation and discover the potential obstacles before designing an adaptative multipath scheduler with advanced metrics. This question is discussed in Papers II and III. We utilize the learning perspective to design the adaptative multipath scheduler in Paper II where we design a functional learning-based multipath scheduler that can adapt within the heterogeneous scenario. In Paper III, we further extend the applicable scenario to generic dynamic networks. We validate the schedulers under various parameter-driven and trace-driven emulated experiments as well as real-world experiments, considering different types of applications.

- Question 3: *How to design and validate an adaptative multipath scheduler that can adapt well and fast simultaneously?*

Based on Questions 1 and 2, we ask this question to meet the central objective of this thesis. This question is partially touched in Paper III and ultimately resolved in Paper IV. Paper IV acknowledges the tradeoff between the adaptation speed and adaptation accuracy and further proposes a solution to alleviate the tradeoff to enhance the adaptation speed and accuracy simultaneously. We validate the scheduler under various parameter-driven and trace-driven emulated experiments and real-world experiments, considering different types of applications.

4 Research Methodology

In this thesis, we aim to solve the above-mentioned research questions from the learning perspective. The research methods conducted in this thesis include *literature review, theoretical analysis*, *ideas development*, *quantitative experiments*, and *data analysis*, presented in sequential order. Among these, *literature review* sets up the knowledge foundation of our work and consolidates the proposed research problems; *theoretical analysis* abstracts the research problem within the corresponding theory framework, easing the understanding of the research problem in a mathematic manner; on the basis of the previous steps, *ideas development* develops the initial hint to the complete solution for the given research problem; *quantitative experiments* validate the proposed solutions in a quantitative manner; and the *data analysis* in return aids us why the proposed solutions work and not work in the given settings, paving the way for future improvements of the proposed solutions.

The detail of each method is described in [59]. Below, we describe how we applied these research methodologies to address the research questions.

Question 1: *What is the status of existing multipath schedulers within the context of modern dynamic networks?*

To answer this question, we review the multipath transport works in the context of modern dynamic networks (*literature review*) and suggest the employment of multipath transport protocols with potential benefits on eMBB and URLLC slices subjecting to the corresponding standardization (*theoretical analysis*), as shown in Paper I. We conclude the lack of adaptative multipath scheduler within such a context.

Question 2: *How to design an adaptative multipath scheduler?*

To answer this question, we start with an initial step to investigate if we can apply the learning aspect within multipath scheduling where we validate the proposed algorithm over a combination of LTE and WLAN in Paper II. In order to guarantee the fast adaptation time and the consideration of the affordability of the real time characteristics, we adopt the light-weight learning rather than the complex deep learning approach. We formulate the multipath scheduling problem and propose a lightweight and deployable online learning solution to this problem (*theoretical analysis, ideas development*). More specifically, given a dynamicity level, a deterministic strategy is derived by using a RL algorithm applied in contextual Multi-Armed Bandit (MAB) scenarios. We further formulate a stochastic adjustment strategy and propose a lightweight derivation and selection of such an adjustment strategy, analyzing its impact on the overall scheduling policy, as a function of the dynamicity levels experienced on the paths (*theoretical analysis, ideas development*). We combine the online learning solution with the stochastic adjustment strategy in the final design of Peekaboo. We demonstrate and analyze the enhanced performance of Peekaboo over the other non-learning based multipath schedulers in both emulations and real-world examinations considering the applications of bulk transfer at different sizes and real-time streaming at different bit rates (*quantitative experiments* and *data analysis*).

Then, we extend the applicable scenario of Peekaboo from dynamic heterogeneous networks to generic dynamic networks in Paper III. To this end, we propose M-Peekaboo. M-Peekaboo extends Peekaboo's learning scheme for path selection toward 5G scenarios that may include paths operating on different frequencies, e.g., mid-band and mmWave by generalizing the action set in Peekaboo to select between all paths, without making any assumptions about the lowest RTT path (*ideas development*). We demonstrate and analyze the superiority of M-Peekaboo over Peekaboo and the other non-learning based multipath schedulers from the 5G-trace-driven emulations across both static and mobile scenarios, interfacing the applications of bulk transfer at different sizes (*quantitative experiments* and *data analysis*).

Question 3: *How to design an adaptative multipath scheduler that can adapt well and fast simultaneously?*

To answer this question, we further improve the learning time and the adaptation accuracy of our learning based multipath scheduler in Paper IV. More specifically, we utilize the combination of online and offline learning, where the online learning performs the fine tuning and the offline learning performs the setup of the templates used for fine tuning. We use meta-learning as the glue to complete the combination of these two approaches and eventually propose FALCON (*theoretical analysis, ideas development*). We demonstrate and analyze the superiority of FALCON over the other learning and non-learning based multipath schedulers from the 5G-trace-driven emulations across both static and mobile scenarios and real-world examinations, considering the applications of bulk transfer and web download at different sizes (*quantitative experiments* and *data analysis*).

For all the papers where we perform experiments, i.e., Paper II, III and IV, we perform the evaluations under the MPQUIC framework, although the scheduling approach theoretically is not restricted to MPQUIC but designed in a generic fashion. For all the papers, we have provided open source software so that the community can replicate our results and further develop the algorithms we have proposed.

5 Related Work and Research Contributions

In this section, we summarize the related work of the thesis from different perspectives including the employment of multipath transport in 5G as presented in Section 5.1, scheduling based on fixed rules as presented in Section 5.2, and scheduling based on learning as presented in Section 5.3. For each perspective, we correspondingly present the research contributions of this thesis.

5.1 Employing Multipath Transport in 5G

There are previous works that survey and summarize the advantages and disadvantages of multipath transport. In [60], a review of load distributing models for multipath networks is provided, with focus on the description of the models rather than the layers where such models can be adopted. The work in [61] provides a review of multipath solutions that specifically solve the reordering problem in heterogeneous wireless networks. Both [62] and [63] survey multipath solutions across different layers. Targeting network-layer multipath solutions, the work in [64] focuses on literature addressing control-plane problems (how to compute and select routes) and data plane problems (how to split flows on the computed paths). A specific aspect of multipath transport protocols, i.e., multipath congestion control, is surveyed in [65]. Nevertheless, none of the aforementioned surveys discuss the application of multipath solutions in 5G, along with the requirements and benefits of doing so. The work in [66] surveys the multipath literature for solutions that can potentially enable URLLC across different layers, including those from 3GPP up to Release 15.

Contributions: The Paper I is the first work to the best of our knowledge that places different multipath transport's works in the context of 5G. Specifically, we map the multipath transport's work to the ATSSS architecture's different steering functional modes, easing their uses in 5G. Different from [66], our work is based on 3GPP Release 16 and consider and suggest multipath transport as the solutions for both eMBB and URLLC.

5.2 Scheduling based on Fixed Rules

Traditional multipath schedulers follow predefined rules that do not change over time. For example, the RR scheduler cyclically sends packets over each path, as long as there is space in the CWND of the paths. RR may perform reasonably well when the available paths have similar characteristics. However, since it does not consider the characteristics of the individual paths it is unable to prevent out-of-order packet arrival at the receiver, which is detrimental to multipath transport performance as presented in Section 2.3.

The minimum RTT (minRTT) scheduler, as the default scheduler in both MPTCP and MPQUIC, prioritizes the paths with available CWND and lowest RTT when sending the data packets. Since minRTT considers path characteristics and exploits faster paths given the underlying congestion control, it has been shown to achieve higher throughput than RR [47].

Blocking Estimation (BLEST) [47] and Earliest Completion First (ECF) [48] try to provide both high throughput and low latency. Assuming two available paths, when both paths have CWND availability, BLEST and ECF behave like minRTT, i.e., they select the path with the lowest RTT. When the path with the lowest RTT has no CWND availability, BLEST and ECF use different mechanisms to decide whether it is less time-consuming to send packets on the path with the highest RTT or wait for the path with the lowest RTT to become available again, where BLEST calculates the time overhead buffered in send window and ECF calculates the time overhead buffered in send buffer.

Out-of-order Transfer for In-order Arrival (OTIAS) [67] is, in many aspects, similar to BLEST and ECF except that OTIAS directly prioritizes the path with the shortest transfer time regardless of whether the path is congested or not. Therefore, the transfer time also includes the time of waiting for the space in the CWND besides the transmission delay. Furthermore, Quality Aware (QAware) [68] is a cross-layer approach incorporating the local queue buffer occupancy information of the Network Interface Card (NIC), aiming at improving the estimation of transmission delay.

Assigning a group of packets to balance the capacity of different paths, Forward Prediction Scheduling (FPS) [69] predicts the packets' arrival time and sends packets in a manner that they are expected to be received in batch. Delay Aware Packet Scheduling (DAPS) [70] directly assigns the number of packets over each path based on the ratio of RTT between the paths. [71] argues that although pre-allocating packets over different paths seem to ensure in-order arrival, there often exists a mismatch between the estimated and the real transfer time, especially in wireless networks. To compensate for the inaccurate estimation, a gap composed of several packets that are not yet scheduled is left between the packets sent over different paths and is self-adjusted based on ACKs which can reflect the out-of-order arrival degree.

Addressing specific use cases and applications, the works in [72], [73], [74], [75] apply packet duplication mechanisms to guarantee robustness, which proves to be effective when extra data usage and battery consumption are not limiting factors. [72] duplicates all the packets, sacrificing the throughput for higher robustness. [73] and [74] adopts the adaptative duplication mechanisms which only duplicate packets when networks are of poor conditions (e.g., of high losses and/or of high variable delays) for the higher expected throughput. Similarly aiming for higher throughput by adaptative duplication, [75] proposes a loss-aware scheduler but solely for networks with more than 20% loss rates.

[49] proposes the Short Transfer Time First (STTF) scheduler which also prioritizes the paths offering shorter transfer time, specifically considering TCP specific aspects such as the TCP Small Queues (TSQ). [76] proposes a multipath scheduler for MPTCP that targets IEEE 802.11 ad/ac WLANs, which continually searches for the optimal ratio of packets sent over the paths.

MP-DASH [77] proposes a scheduling framework for video streaming that is aware of network interface preferences from users, e.g., prioritizing WiFi over cellular links. The scheduling decision is deducted by solving an integer programming problem to minimize the usage of the unwanted path while trying to meet users' Quality of Experience (QoE) requirements.

The work [78] adopts the purchased price of the path as the prior information. It is assumed that, under a guaranteed throughput, the users prefer to use the path having lower costs. Then, by applying Lyapunov optimization, the work aims at maximizing the throughput while minimizing the price cost for users. Also adopting the path cost to derive the priority, [79] proposes a cost-based scheduling algorithm, which simultaneously reduces the cost of multipath use for network operators and retains the QoE levels required by the end-users in case of bursty video-on-demand traffic.

Contributions: Schedulers based on predefined rules can adapt fast but not accurately to time-varying network conditions. This is due to the inherent limitation caused by predefining the rule to follow for scheduling packets over the available paths. Indeed, the rule is usually rather simple and coarse-grained (e.g., select the path with minimum average RTT), thus failing to adapt accurately to the complex dynamics of the network conditions. To solve this problem, Paper II proposes a learning based multipath scheduling approach for heterogeneous networks and Paper III further extends the applicable scenarios to generic networks. The results shown in Papers II and III demonstrate the superiority of applying learning based scheduling approaches over the fixed rules ones. Paper IV further improves the performance over the schedulers proposed in Paper II and III, as well as the ones presented as fixed rules.

5.3 Scheduling based on Learning

During the writing of this thesis, we could not find many works that targets at designing learning-based multipath scheduling. To ensure the theoretical completeness of our work, we expand our related work to other networking systems that exploit learning concept in their design.

Scheduling based on Online Learning

This paradigm assumes that to derive a model and/or policy, an ML algorithm uses data that is collected while the model/policy is being derived and used. In the following, we refer to run-time collected data as online data. Differently from the offline learning paradigm, the learning outcome is thus modified and adapted at run-time, exploiting newly encountered environment characteristics, i.e., new online data. This is commonly performed via two main approaches, i.e., with or without the use of an abandoning mechanism.

In the first approach, the model/policy is abandoned when either a significant change in the environment characteristics is detected via so-called change point detection (i.e., detect if the current ongoing context is the same with the training context), e.g., as used for mobile network diagnostics [80], or a predefined timer expires, e.g., as used for congestion control where the gradient-based approach is employed to search for the optimal sending rate at each timer period [81] [82] and as used for the optimization of Quality of Experience (QoE) where the Discounted Upper-Confidence Bound (UCB) [83] is employed to make the past experience expired [84]. Indeed, if some environment characteristics reappear after being forgotten, the paradigm needs to derive again the model/policy that better suits such characteristics.

In the second approach, the online learning algorithm does not apply the abandoning mechanism, i.e., the model/policy is continuously updated since the algorithm is continuously fed with online data, as shown in [85] for congestion control. Hence, in this case, there is no abrupt model/policy abandoning, which may cause a slower reaction to sudden changes in the environment characteristics. Examples of multipath schedulers that use online learning with no abandoning mechanisms are [86], [87]. Across [85] [86] [87], the similarity is to utilize different variants of Q-learning within Reinforcement learning [88] to derive either the congestion control policy or the scheduling policy, where the objective is to select the best action (e.g., the sending rate in congestion control and the path to transmit the packet in multipath scheduling) given current state information (e.g., bandwidth, delay, and loss conditions) for a high cumulative reward (e.g., high throughput or low transfer time), suppose the scenario is subject to the modeling of Markov Decision Process (MDP).

With the abandoning mechanism (the first approach), old models/policies that trained to be optimal for specific environment characteristics may thus be discarded and accordingly adapt for the current environment characteristics; without the abandoning mechanism (the second approach), the continuous feed of online data may result in the slow adaptation for the current environment characteristics while also cannot efficiently keep the memory of the previous models (aka. the well-known catastrophic forgetting problem [89]). As a partial remedy for the second approach, [90] tries to apply lifelong learning to alleviate the catastrophic forgetting problem.

Contributions: Schedulers based on online learning can ensure the derivation of an accurate scheduling policy. In general, however, the need for learning the network conditions online makes the adaptation slower compared to schedulers based on predefined rules. If the adaptation time is too long, the multipath scheduler might follow a policy that is not functional at all. In order to speed up adaptation, schedulers based on online learning can sacrifice accuracy, thus exploiting a limited amount of data and a simple learning architecture for deriving a policy. In Papers II and III, we explore this trajectory and design the multipath scheduler that are functional and provide superior performance over the state-of-the-arts. In addition, Papers II and III both include the change point detection into its design for the obvious advantages as mentioned above. Further, as proposed in Paper IV, we utilize the offline learning to bootstrap the online learning thus can achieving fast adaptation speed and ensure the complexity of the mode without trading off the complexity of the model.

Scheduling based on Offline Learning

This paradigm assumes that, in order to derive a model of and/or a policy for a generic environment, an ML algorithm uses environment characteristics, i.e., data, collected well-ahead, before the derived model is meant to be used. In the following, we refer to pre-collected data as offline data. The learning outcome, e.g., the policy to be used by a network protocol, is not modified once derived on offline data. In other words, there is no retraining. Therefore, the assumption is that offline data includes a complete enough set of environment characteristics that could be experienced when the model/policy is actually used.

To the best of our knowledge, there is no existing multipath scheduling approach based on offline learning and we analyze the feasibilities in the later part of this subsection. Within the other context of the networking field, offline learning is used to derive offline data-based policies for congestion control [91], Adaptive Bit Rate (ABR) streaming [92] [93], resource management of the data center [94], and resource management of the mobile device [95].

In [91], simulated environments comprised of different settings are set up to mimic different scenarios that the congestion controller faces in the real world. The optimal parameter of controller policy (i.e., how to update the congestion window and regulate the lower bound of time between two successive sends) is searched to maximize the objective function that prioritizes throughput and delay for each setting in the simulated environment and eventually obtains a set of control rules. In [92] [93], the simulated video streaming environment is set up over the collected network traces. Traversing over the simulated environment, [92] and [93] use different reinforcement learning algorithms (i.e., Deep Q-Network (DQN) [96] and Asynchronous Advantage Actor Critic (A3C) [97], respectively) to train the policy. Using similar learning methodologies, [94] uses the DQN to train the workload scheduling approach for the data center cluster over the simulated environment set up over the workload trace. [95] treats the task of scheduling webpage rendering on heterogeneous cores as a classification problem. Given the webpage and hardware information as the feature, the optimal core to use is the class that the feature represents. [95] utilizes Support Vector Machine (SVM) [98] to train over the collected workload trace to obtain the scheduling algorithm.

Contributions: Schedulers based on offline learning may intuitively seem like a reasonable approach for achieving both fast and accurate adaptation. An offline learning-based scheduler may adapt fast because it is pre-trained. Moreover, such a scheduler might achieve accurate adaptation if trained on all the possibly encountered network conditions. However, this assumption is rather unrealistic for two main reasons: (1) Collecting all possible network conditions (past and future) is nearly impossible; (2) Even if all combinations of network conditions could be found, it is difficult to accurately label each of them mathematically. Hence, several combinations of network conditions may be involved in the pre-training, and the obtained model would still have a coarse-grained match with the fine-grained network conditions. We show in Paper IV that the approaches proposed in Paper III also outperforms the offline learning based approach across the examined practical scenarios due to the ability to adpat online. Further, the approach proposed in Paper IV itself combines the advantages of online learning with offline learning as mentioned above.

6 Summary

In this section, we conclude our work in Section 6.1 and present the future work based on the limitations of our work in Section 6.2.

6.1 Conclusions

This thesis focuses on the design and validation of adaptive learning-based multipath schedulers for 5G networks and beyond. The research work is motivated by schedulers based on fixed rules having difficulty tackling existing and ever-increasing degrees of dynamicity in these networks. The dynamicity presents as the variation of bandwidth, delay, and loss, which can be caused by the employment of wireless networks, especially the mmWave frequency channels, and congestion in the end-to-end network, etc.

To design an adaptive multipath scheduler for 5G networks and beyond, we first review the status of multipath transport as a whole within 5G. Therein, we acknowledge the need to design the adaptative multipath scheduler and propose utilizing the learning-based design as a potential solution.

Next, we consider a subproblem where the paths are heterogeneous with dynamically changing path characteristics. To this end, we propose Peekaboo, an adaptive multipath scheduler that leverages an online learning mechanism in combination with a stochastic adjustment strategy to adapt to the dynamic characteristics of the paths. Peekaboo is computationally lightweight and easily deployable. We implement Peekaboo in MPQUIC and compare its performance with state-of-the-art multipath schedulers for a wide range of dynamicity levels, using both emulated networks and real network scenarios. Across the examined scenarios and applications, Peekaboo consistently offers superior or similar performance to the multipath schedulers based on fixed rules.

Then, we extend the applicable scenarios of Peekaboo to generic dynamic networks. To this end, we propose M-Peekaboo on the basis of Peekaboo with a change of learning framework in terms of the action set. We validate different schedulers over the ATSSS like architecture with the combination of links of 4G, 5G, and WLAN across both static and mobile networks. More specifically, we employ real-world traces for the validations. M-Peekaboo is shown to outperform both the Peekaboo and multipath schedulers based on fixed rules.

Nevertheless, both M-Peekaboo and Peekaboo are online learning based schedulers and they are shown to adapt not fast enough in rapidly changing. To overcome this drawback, we propose FALCON, a learning-based multipath scheduler that can adapt fast and accurately to changing network conditions by combining the benefits of online and offline learning. Through extensive emulations, we show that FALCON is able to consistently outperform all state-of-the-art schedulers by adapting to the network conditions in a fast and accurate manner. Our real-world experiments confirm that FALCON performs well also under realistic network settings.

6.2 Limitations and Future work

We present limitations for this work and correspondingly motivate future directions.

From the learning perspective, we have demonstrated the possibility of applying DQN within our final solution, FALCON, but we could also consider applying other deep learning approaches to enhance the scheduler's performance. Further, the depth of the neural network is usually in a positive correlation with the learning model’s performance, thus impacting the scheduler’ s performance. Nevertheless, the increase of the depth also increases the training time of the model for fixed hardware, ultimately impacting the scheduler’s adaptation time. Depending on the design goal, the relationship among the cost of hardware (including the financial aspect and the power aspect), the depth of the neural network, and the adaptation time can be exploited. Lastly, the outcome of the learning remains as a black box for us, it would be beneficial to interpret and understand the learning outcome to better control its behavior.

From the networking perspective, when looking upwards from the scheduling, in the application layer, we tested over several application types including bulk transfer, web download, and real-time streaming in this work. One of the attractive spots of 5G is that it may interface various types of applications, such as tactile internet, VR/AR, whose manner of operation might be different from the classical applications. It is therefore particularly interesting to extends our work to support these 5G applications. When looking downwards from scheduling, in the transport layer, we can think of co-designing the multipath scheduler and multipath congestion control. As such, for example, the multipath congestion control’s mechanism is no longer a hidden state for the scheduler’s MDP, potentially enhancing its performance.

Bibliography

[1] L. Kleinrock, "An Early History of the Internet," *IEEE Communications Magazine,* vol. 48, no. 8, pp. 26-36, 2010.

[2] X. Lin et al., "5G New Radio: Unveiling the Essentials of the Next Generation Wireless Access Technology," *IEEE Commun. Standards Mag.*, vol. 3, no. 3, pp. 30-37, 2019.

[3] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong and J. C. Zhang, "What will 5G be?," *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065-1082, 2014.

[4] M. Agiwal, A. Roy and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 3, pp. 1617-1655, 2016.

[5] 3GPP, "TS 23.501: System Architecture for the 5G System," 3GPP, 2020.

[6] H. Wu, G. Caso, S. Ferlin, Ö. Alay and A. Brunstrom , "Multipath Scheduling for 5G Networks: Evaluation and Outlook," *IEEE Communications Magazine,* vol. 59, no. 4, pp. 44-50, 2021.

[7] M. Series, "IMT Vision--Framework and overall objectives of the future development of IMT for 2020 and beyond," *Recommendation ITU,* pp. 2083-0, 2015.

[8] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega and D. Bega et al., "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Communications magazine,* vol. 55, no. 5, pp. 72-79, 2017.

[9] "NR; Physical layer procedures for control," *3GPP TS 38.213 v16.2.0,* 2020.

[10] A. Arjun, G. d. Veciana and S. Shakkottai, "Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018.

[11] A. K. Bairagi, M. S. Munir, M. Alsenwi, N. H. Tran, S. S. Alshamrani, M. Masud, Z. Han and C. S. Hong, "Coexistence Mechanism between eMBB and uRLLC in 5G Wireless Networks," *IEEE Transactions on Communications*, 2020.

[12] 3GPP, "TS 24.302: Access to the Evolved Packet Core (EPC) via non-3GPP access networks”," 3GPP, 2009.

[13] 3GPP, "TS 36.300: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2 (Release 11)," 3GPP, 2015.

[14] 3GPP, "TS 36.300: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2 (Release 12)," 3GPP, 2016.

[15] 3GPP, "TS 36.300: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Multi-connectivity; Stage 2 (Release 15); V15.3.0," 3GPP, 2018.

[16] 3GPP, "TS 36.361: LTE/WLAN Radio Level Integration Using IPsec Tunnel (LWIP) encapsulation Protocol specification, Release 14," 3GPP, 2017.

[17] D. Laselva, D. Lopez-Perez, M. Rinne and T. Henttonen, "3GPP LTE-WLAN aggregation technologies: Functionalities and performance comparison," *IEEE Communications Magazine,* vol. 56, no. 3, pp. 195-203, 2018.

[18] R. Bajracharya, R. Shrestha, R. Ali, A. Musaddiq and S. W. Kim, "LWA in 5G: State-of-the-art architecture, opportunities, and research challenges," *IEEE Communications Magazine,* vol. 56, no. 10, pp. 134-141, 2018.

[19] 3GPP, "TS 37.340: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Multi-connectivity; Stage 2 (Release 15); V15.3.0," 3GPP, 2018.

[20] J. R. Iyengar, P. D. Amer and R. Stewart, "Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths," *IEEE/ACM Transactions on Networking(ToN),* vol. 14, no. 5, pp. 951-964, 2006.

[21] D. Wischik, C. Raiciu, A. Greenhalgh and M. Handley, "Design, Implementation and Evaluation of Congestion Control for Multipath TCP," in *NSDI*, 2011.

[22] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe and R. Steinmetz, "Multipath QUIC: A deployable multipath transport protocol," in *2018 IEEE International Conference on Communications (ICC)*, 2018.

[23] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili and D. Towsley, "A measurement-based study of multipath TCP performance over wireless networks," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013.

[24] S. Deng, R. Netravali, A. Sivaraman and H. Balakrishnan, "Wifi, LTE, or Both?: Measuring Multi-homed Wireless Internet Performance," in *ACM IMC 2014*, 2014.

[25] C. Paasch, G. Detal, F. Duchene, C. Raiciu and O. Bonaventure, "Exploring mobile/WiFi handover with multipath TCP," in *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, 2012.

[26] Q. De Coninck, M. Baerts, B. Hesmans and O. Bonaventure, "A First Analysis of Multipath TCP on Smartphones," in *Passive and Active Measurement*, 2016.

[27] 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, 2016.

[28] Q. De Coninck and O. Bonaventure, "MultipathTester: Comparing MPTCP and MPQUIC in Mobile Environments," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*, 2019.

[29] F. Fejes, S. Rácz and G. Szabó, "Application agnostic QoE triggered multipath switching for Android devices," in *2017 IEEE International Conference on Communications (ICC)*, 2017.

[30] S. K. Saha, A. Kannan, G. Lee, N. Ravichandran, P. K. Medhe, N. Merchant and D. Koutsonikolas, "Multipath TCP in Smartphones: Impact on Performance, Energy, and CPU Utilization," in *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*, 2017.

[31] "Improving Network Reliability Using Multipath TCP," [Online]. Available: https://developer.apple.com/documentation/foundation/urlsessionconfiguration/improving_network_reliability_using_multipath_tcp.

[32] Q. An, Y. Liu, Y. Ma and Z. Li, "Multipath Extension for QUIC," October 2020. [Online]. Available: http://www.ietf.org/internet-drafts/draft-an-multipath-quic-00.txt.

[33] J. Deutschmann, K.-S. Hielscher and R. German, "Multipath Communication with Satellite and Terrestrial Links," October 2020. [Online]. Available: http://www.ietf.org/internet-drafts/draft-deutschmann-sat-ter-multipath-00.txt.

[34] N. Keukeleire and B. Hesmans, "Increasing Broadband Reach with Hybrid Access Networks," *IEEE Communications Standards Magazine,* vol. 4, no. 1, pp. 43-49, 2020.

[35] J. R. Iyengar, P. D. Amer and R. Stewart, "Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths," *IEEE/ACM Transactions on Networking(ToN),* vol. 14, no. 5, pp. 951-964, 2006.

[36] J. Iyengar, K. Shah, P. Amer and R. Stewart, "Concurrent multipath transfer using SCTP multihoming," in *SPECTS 2004*, 2004.

[37] A. Ford, C. Raiciu, M. Handley, O. Bonaventure and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses," March 2020. [Online]. Available: http://www.rfc-editor.org/rfc/rfc8684.txt.

[38] Q. Coninck and O. Bonaventure, "Multipath Extensions for QUIC (MP-QUIC)," November 2020. [Online]. Available: http://www.ietf.org/internet-drafts/draft-deconinck-quic-multipath-06.txt.

[39] Q. An, Y. Liu, Y. Ma and Z. Li, "Multipath Extension for QUIC," October 2020. [Online]. Available: https://tools.ietf.org/html/draft-an-multipath-quic-00.

[40] M. Amend, D. Hugo, A. Brunstrom, A. Kassler, V. Rakocevic and S. Johnson, "DCCP Extensions for Multipath Operation with Multiple Addresses," 08 2021. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tsvwg-multipath-dccp/00/.

[41] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure and M. Handley, "How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP," in *USENIX NSDI 2012*, 2012.

[42] S. R. Das, "Evaluation of QUIC on web page performance," Massachusetts Institute of Technology, 2014.

[43] K. L. McMillan and L. D. Zuck, "Formal specification and testing of QUIC," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019.

[44] O. Bonaventure, M. Piraux, Q. Coninck, M. Baerts, C. Paasch and M. Amend, "Multipath schedulers," March 2020. [Online]. Available: http://www.ietf.org/internet-drafts/draft-bonaventure-iccrg-schedulers-00.txt.

[45] S. Ferlin, T. Dreibholz and Ö. Alay, "Multi-path transport over heterogeneous wireless networks: Does it really pay off?," in *IEEE Global Communications Conference (GLOBECOM)*, 2014.

[46] C. Paasch, S. Ferlin, Ö. Alay and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," in *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, 2014.

[47] S. Ferlin, Ö. Alay, O. Mehani and R. Boreli, "BLEST: Blocking Estimation-based MPTCP Scheduler for Heterogeneous Networks," in *IFIP Networking*, 2016.

[48] Y.-s. Lim, E. M. Nahum, D. Towsley and R. J. Gibbens, "ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths," in *ACM CoNEXT*, 2017.

[49] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, O. Alay and N. Kuhn, "Low-Latency Scheduling in MPTCP," in *IEEE/ACM Transactions on Networking(ToN)*, 2019.

[50] 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)*, 2012.

[51] C. Raiciu, M. Handley and D. Wischik, "Coupled congestion control for multipath transport protocols," IETF RFC 6356, Oct, 2011.

[52] C. Raiciu, D. Wischik and M. Handley, "Practical congestion control for multipath transport protocols," University College London, London/United Kingdom, Tech. Rep, 2009.

[53] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe and R. Steinmetz, "Multipath QUIC: A deployable multipath transport protocol," in *2018 IEEE International Conference on Communications (ICC)*, 2018.

[54] I. Johansson, "Congestion control for 4G and 5G access," 2016. [Online]. Available: https: //tools.ietf.org/html/draft-johansson-cc-for-4g-5g-02.

[55] B. Adamson, C. Adjih, J. Bilbao, V. Firoiu, F. Fitzek, S. Ghanem, E. Lochin, A. Masucci, M.-J. Montpetit, M. Pedersen, G. Peralta, V. Roca, P. Saxena and S. Sivakumar, "Taxonomy of Coding Techniques for Efficient Network Communications," RFC Editor, 2018.

[56] N. Kuhn, E. Lochin, F. Michel and M. Welzl, "Coding and congestion control in transport," March 2020. [Online]. Available: http://www.ietf.org/internet-drafts/draft-irtf-nwcrg-coding-and-congestion-02.txt.

[57] S. Ferlin, S. Kucera, H. Claussen and Ö. Alay, "MPTCP Meets FEC: Supporting Latency-Sensitive Applications Over Heterogeneous Networks," *IEEE/ACM Transactions on Networking,* no. 99, pp. 1-14, 2018.

[58] P. Saxena, T. Dreibholz, H. Skinnemoen, Ö. Alay, A. Vazquez-Castro, S. Ferlin and G. Acar, "Resilient Hybrid SatCom and Terrestrial Networking for Unmanned Aerial Vehicles," in *Proceedings of the 39th IEEE International Conference on Computer Communications (INFOCOM), International Workshop on Wireless Sensor, Robot and UAV Networks (WiSARN)*, 2020.

[59] R. Jain, The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling, John Wiley & Sons, 1990.

[60] S. Prabhavat, H. Nishiyama, N. Ansari and N. Kato, "On load distribution over multipath networks," *IEEE Communications Surveys & Tutorials,* vol. 14, no. 3, pp. 662-680, 2011.

[61] A. L. Ramaboli, O. E. Falowo and A. H. Chan, "Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues," *Journal of Network and Computer Applications,* vol. 35, no. 6, pp. 1674-1690, 2012.

[62] S. K. Singh, T. Das and A. Jukan, "A survey on internet multipath routing and provisioning," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 4, pp. 2157-2175, 2015.

[63] M. Li, A. Lukyanenko, Z. Ou, A. Ylä-Jääski, S. Tarkoma, M. Coudron and S. Secci, "Multipath transmission for the internet: A survey," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 4, pp. 2887-2925, 2016.

[64] J. Qadir, A. Ali, K.-L. A. Yau, A. Sathiaseelan and J. Crowcroft, "Exploiting the power of multiplicity: a holistic survey of network-layer multipath," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 4, pp. 2176-2213, 2015.

[65] 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, 2016.

[66] M.-T. Suer, C. Thein, H. Tchouankem and L. Wolf, "Multi-Connectivity as an Enabler for Reliable Low Latency Communications-An Overview," *IEEE Communications Surveys & Tutorials,* 2019.

[67] F. Yang, Q. Wang and P. D. Amer, "Out-of-order transmission for in-order arrival scheduling for multipath TCP," in *IEEE 28th International Conference on Advanced Information Networking and Applications Workshops,* 2014.

[68] T. Shreedhar, N. Mohan, S. K. Kaul and J. Kangasharju, "QAware: A Cross-Layer Approach to MPTCP Scheduling," in *IFIP Networking*, 2018.

[69] F. H. Mirani, M. Kherraz and N. Boukhatem, "Forward prediction scheduling: Implementation and performance evaluation," in *IEEE 18th International Conference on Telecommunications*, 2011.

[70] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani and R. Boreli, "DAPS: Intelligent delay-aware packet scheduling for multipath transport," in *IEEE ICC*, 2014.

[71] H. Shi, Y. Cui, X. Wang, Y. Hu, M. Dai, F. Wang and K. Zheng, "STMS: Improving MPTCP Throughput Under Heterogeneous Networks," in *USENIX ATC*, 2018.

[72] A. Frommgen, T. Erbshäußer, A. Buchmann, T. Zimmermann and K. Wehrle, "ReMP TCP: Low latency Multipath TCP," in *IEEE ICC* , 2016.

[73] H. Lee, J. Flinn and B. Tonshal, "Raven: Improving interactive latency for the connected car," in *ACM MobiCom*, 2018.

[74] Y. E. Guo, A. Nikravesh, Z. M. Mao, F. Qian and S. Sen, "Accelerating Multipath Transport Through Balanced Subflow Completion," in *ACM MobiCom*, 2017.

[75] E. Dong, M. Xu, X. Fu and Y. Cao, "A Loss Aware MPTCP Scheduler for Highly Lossy Networks," *Computer Networks,* 2019.

[76] S. K. Saha, S. Aggarwal, R. Pathak, D. Koutsonikolas and J. Widmer, "MuSher: An Agile Multipath-TCP Scheduler for Dual-Band 802.11 ad/ac Wireless LANs," in *ACM Mobicom*, 2019.

[77] B. Han, F. Qian, L. Ji and V. Gopalakrishnan, "MP-DASH: Adaptive video streaming over preference-aware multipath," in *ACM Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies,* 2016.

[78] K. Gao, C. Xu, J. Qin, L. Zhong and G.-M. Muntean, "A Stochastic Optimal Scheduler for Multipath TCP in Software Defined Wireless Network," in *IEEE ICC,* 2019.

[79] M. Amend, V. Rakocevic and J. Habermann, "Cost optimized multipath scheduling in 5G for Video-on-Demand traffic," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2021.

[80] A. Padmanabha Iyer, L. Erran Li, M. Chowdhury and I. Stoica, "Mitigating the latency-accuracy trade-off in mobile data analytics systems," in *ACM Mobicom*, 2018.

[81] M. Dong, T. Meng, D. Zarchy, E. Arslan, Y. Gilad, B. Godfrey and M. Schapira, "PCC vivace: Online-learning congestion control," in *USENIX NSDI*, 2018.

[82] T. Gilad, N. Rozen-Schiff, P. B. Godfrey, C. Raiciu and M. Schapira, "MPCC: online learning multipath transport," in *ACM CoNEXT*, 2020.

[83] A. Garivier and E. Moulines, "On upper-confidence bound policies for non-stationary bandit problems," *arXiv preprint arXiv:0805.3415,* 2008.

[84] J. Jiang, S. Sun, V. Sekar and H. Zhang, "Pytheas: Enabling data-driven quality of experience optimization using group-based exploration-exploitation," in *USENIX NSDI*, 2017.

[85] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang and S. Lu, "SmartCC: A Reinforcement Learning Approach for Multipath TCP Congestion Control in Heterogeneous Networks," in *IEEE Journal on Selected Areas in Communications*, 2019.

[86] M. M. Roselló, "Multi-path Scheduling with Deep Reinforcement Learning," in *2019 European Conference on Networks and Communications (EuCNC)*, 2019.

[87] H. Zhang, W. Li, S. Gao, X. Wang and B. Ye, "ReLeS: A Neural Adaptive Multipath Scheduler based on Deep Reinforcement Learning," in *IEEE INFOCOM*, 2019.

[88] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[89] R. Kemker, M. McClure, A. Abitino, T. L. Hayes and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *AAAI*, 2018.

[90] T. Huang, C. Zhou, X. Yao, R.-X. Zhang, C. Wu, B. Yu and L. Sun, "Quality-aware neural adaptive video streaming with lifelong imitation learning," *IEEE Journal on Selected Areas in Communications,* vol. 38, no. 10, pp. 2324-2342, 2020.

[91] K. Winstein and H. Balakrishnan, "Tcp ex machina: Computer-generated congestion control," in *ACM SIGCOMM Computer Communication Review*, 2013.

[92] Z. Akhtar, Y. S. Nam, R. Govindan, S. Rao, J. Chen, E. Katz-Bassett, B. Ribeiro, J. Zhan and H. Zhang, "Oboe: auto-tuning video ABR algorithms to network conditions," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018.

[93] H. Mao, R. Netravali and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017.

[94] H. Mao, M. Schwarzkopf, S. B. Venkatakrishnan, Z. Meng and M. Alizadeh, "Learning scheduling algorithms for data processing clusters," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019.

[95] J. Ren, X. Wang, J. Fang, Y. Feng, D. Zhu, Z. Luo, J. Zheng and Z. Wang, "Proteus: network-aware web browsing on heterogeneous mobile systems," in *ACM CoNEXT*, 2018.

[96] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland and G. Ostrovski, "Human-level control through deep reinforcement learning," *nature,* vol. 518, no. 7540, pp. 529-533, 2015.

[97] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016.

[98] W. S. Noble, "What is a support vector machine?," *Nature biotechnology,* vol. 24, no. 12, pp. 1565-1567, 2006.

Article I:

Wu, Hongjia, Simone Ferlin, Giuseppe Caso, Özgü Alay, and Anna Brunstrom. "A Survey on Multipath Transport Protocols Towards 5G Access Traffic Steering, Switching and Splitting." *IEEE Access 9 (2021): 164417-164439.*

DOI: **10.1109/ACCESS.2021.3134261**

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 6, 2021, accepted November 29, 2021, date of publication December 10, 2021,
date of current version December 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3134261

A Survey on Multipath Transport Protocols Towards 5G Access Traffic Steering, Switching and Splitting

HONGJIA WU[1], SIMONE FERLIN[2], GIUSEPPE CASO[3], (Member, IEEE), ÖZGÜ ALAY[4,5], (Member, IEEE), AND ANNA BRUNSTROM[6], (Member, IEEE)

[1]SimulaMet and OsloMet, 0167 Oslo, Norway
[2]Ericsson AB, 164 40 Stockholm, Sweden
[3]Ericsson Research, Radio Systems and Standards, 164 40 Stockholm, Sweden
[4]Department of Informatics, University of Oslo, 0316 Oslo, Norway
[5]SimulaMet, 0167 Oslo, Norway
[6]Department of Computer Science, Karlstad University, 651 88 Karlstad, Sweden

Corresponding author: Hongjia Wu (hongjia@simula.no)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program, 5th Generation End-to-end Network, Experimentation, System Integration, and Showcasing (5GENESIS) under Grant 815178.

ABSTRACT The fifth generation (5G) cellular network aims at providing very high data rates, ultra reliable low latency communications, and a vast increase of connection density. As one of the design trends towards these objectives, 5G exploits multi-connectivity, i.e., the concurrent use of multiple access networks. The Access Traffic Steering, Switching, and Splitting (ATSSS) architecture has recently been proposed to enable 5G multi-connectivity, and multipath transport protocols have emerged as a key ATSSS technology enabler. Within this context, this survey presents a detailed review of multipath transport protocols, identifies their existing and potential exploitation in ATSSS, and suggests their applicability for enhanced Mobile Broadband (eMBB) and Ultra Reliable Low Latency Communications (URLLC) services. To this end, we first review 5G background and current standardization activities around multi-connectivity and the ATSSS architecture. We then provide an in-depth review of multipath transport protocols, covering four core functionalities, i.e., path management, scheduling, congestion control, and reliable transfer. Based on the reviewed literature, we further discuss the integration of multipath transport into ATSSS to achieve eMBB and URLLC service requirements. Finally, we also point out major open research issues and discuss possible future directions.

INDEX TERMS Multipath transport protocols, access traffic steering, switching and splitting (ATSSS), enhanced mobile broadband (eMBB), ultra reliable low latency communication (URLLC).

I. INTRODUCTION

The 5th generation of mobile communications (5G) raises the expectations towards connecting the whole society and exploits multiple technologies to be able to accommodate the requirements of a wide range of services. As defined by the International Telecommunication Union (ITU), three major performance aspects are central in 5G: very high data rates, ultra-reliable and low latency, and massive connectivity. As such, the ITU classifies 5G services into three main categories: enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine Type Communications (mMTC). eMBB aims to meet the people's demand for an increasingly digitally connected lifestyle and focuses on services that have high bandwidth requirements such as high definition (HD) video streaming, and virtual/augmented reality (VR/AR) applications. URLLC aims to meet digital industry expectations and focuses on latency-sensitive and high-reliability services such as assisted and automated driving, remote robotics, and mission-critical applications. mMTC aims to meet demands for a fully-connected digital society, focusing on services that include high connection density requirements such as smart cities and smart agriculture [1]. Figure 1 illustrates some examples of envisioned use cases for these three categories, representing the topological relationship in a triangle.

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Maaz Rehan.

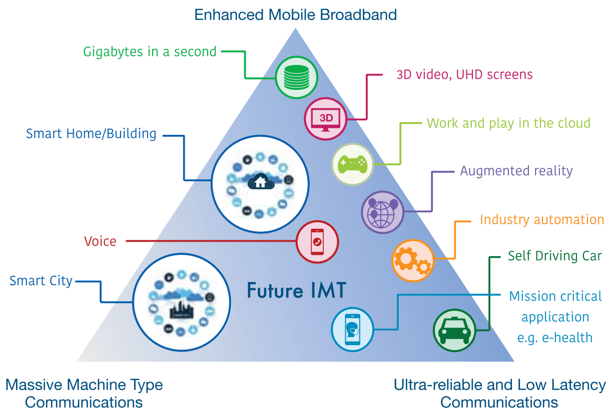


FIGURE 1. **5G services and corresponding reference use cases [2].**

To fulfill the requirements of these use cases, several enhancements have been proposed both in radio access and core networks [10]. Among others, millimeter wave (mmWave), Massive MIMO, Network Slicing, Software-defined Networking (SDN), Network Function Virtualization (NFV), and Multi-access Edge Computing (MEC), significantly contribute to shaping the 5G architecture [11], [12]. Further, the 5G services highlight the need for *multi-connectivity* in order to meet the aforementioned requirements. By exploiting multiple Radio Access Technologies (RATs) simultaneously [11], multi-connectivity provides not only Quality of Service (QoS) improvements on the user side, but also better load balancing across available RATs on the network side.

Over the years, many schemes and methods for enabling efficient and reliable multi-connectivity have been proposed, especially at the radio level. Transport layer approaches have recently gained significant attention due to the Technical Specification (TS) 23.501 (Release 16) [13] by 3rd Generation Partnership Project (3GPP), which specifies how the 5G system can be extended to support Access Traffic Steering, Switching and Splitting (ATSSS) between 3GPP access (e.g., LTE and 5G New Radio (NR)) and non-3GPP access networks (e.g., WiFi). ATSSS leverages multipath transport protocols to deliver the functionalities by manipulating traffic at the flow or intra-flow level [13], [14]. By doing so, ATSSS can conform to eMBB requirements, delivering increased throughput through concurrent transmissions, and to URLLC requirements, delivering low latency and high reliability through path redundancy.

Motivated by the benefits that ATSSS and multipath transport protocols can bring to eMBB and URLLC services, this paper surveys the state-of-the-art research efforts on multipath transport protocols, identifies how they can be leveraged in ATSSS, and suggests which 5G requirements they help to meet.

A. RELATED SURVEYS

Putting our work in context, Table 1 lists related surveys on multipath transmission. In [3], a review of load distributing models for multipath networks is provided, with focus on the description of the models rather than the layers where such models can be adopted. The work in [4] provides a review of multipath solutions that specifically solve the reordering problem in heterogeneous wireless networks. Both [5] and [7] survey multipath solutions across different layers. Targeting network-layer multipath solutions, the work in [6] focuses on literature addressing control-plane problems (how to compute and select routes) and data plane problems (how to split flows on the computed paths). A specific aspect of multipath transport protocols, i.e., multipath congestion control, is surveyed in [8]. Nevertheless, none of the aforementioned surveys discuss the application of multipath solutions in 5G, along with the requirements and benefits of doing so.

The work in [9] surveys the multipath literature for solutions that can potentially enable URLLC across different layers, including those from 3GPP up to Release 15. Differently from [9], our work is based on 3GPP Release 16,[1] and it specifically surveys multipath literature focusing on transport layer solutions, due to their direct applicability in ATSSS. Secondly, to ease the link between multipath transport and 5G, we present the solutions from the multipath literature and position them in relation to the ATSSS *steering modes*, introduced in Section II-E. Thirdly, our work surveys the multipath literature addressing both eMBB and URLLC services.

B. CONTRIBUTIONS AND OUTLINE OF THIS SURVEY

The main contribution of this survey can be summarised as follows:

- We provide an overview of 5G services and their requirements with a link to multi-connectivity approaches meant to address such requirements;
- We focus on multi-connectivity solutions at the transport layer, and thus analyze the main functional blocks of multipath transport protocols, i.e., path management, scheduling, congestion control, and reliable transfer;
- We describe the two main options for integrating multipath transport protocols in 5G systems, i.e., *above-the-core* and *core-centric*. For the second case, we particularly analyze the ATSSS architecture, as the most recent multi-connectivity mechanism standardized by 3GPP;
- We provide a comprehensive review of work related to multipath transport, and discuss how the main components of multipath transport protocols map to specific ATSSS functionalities and modes;
- We identify and discuss open research issues in ATSSS and multipath transport in 5G.

To present our contributions, we outline the survey as reported in Figure 2: In Section II we review 5G background

[1] We base our work in this survey on ATSSS's Rel-16 [13], [15], which lay the foundations for multi-connectivity in 5G. Currently, 3GPP's ATSSS Rel-17 (Phase 2) [16] is ongoing work and expected to be concluded during 2022. Several proposals in this phase focus on the impact in the 5G and beyond architecture as well as extensions of ATSSS. While we briefly mention the ongoing standardization efforts in Section II-E, we opt to not heavily rely on them at this stage.

TABLE 1. Overview of related work and comparison with the present contribution.

Reference	Year	Focus of the survey	OSI layer	5G aspects
[3]	2011	Generalized load distributing models in multipath scheduling	N/A	N/A
[4]	2012	Multipath solutions over reordering problem in heterogeneous wireless networks	Link, network, transport, application	N/A
[5]	2015	Multipath solutions at different layers	Physical, link, network, transport, application	N/A
[6]	2015	Multipath solutions at network layer	Network	N/A
[7]	2016	Multipath solutions at different layers	Link, network, transport, application	N/A
[8]	2016	Multipath congestion control	Transport	N/A
[9]	2019	Multipath solutions for URLLC	Link, network, transport, application	Overview up to 3GPP Rel-15
This work	2021	Multipath transport protocols for eMBB and URLLC	Transport	Focus on 3GPP Rel-16; ATSSS analysis and mapping with multipath transport

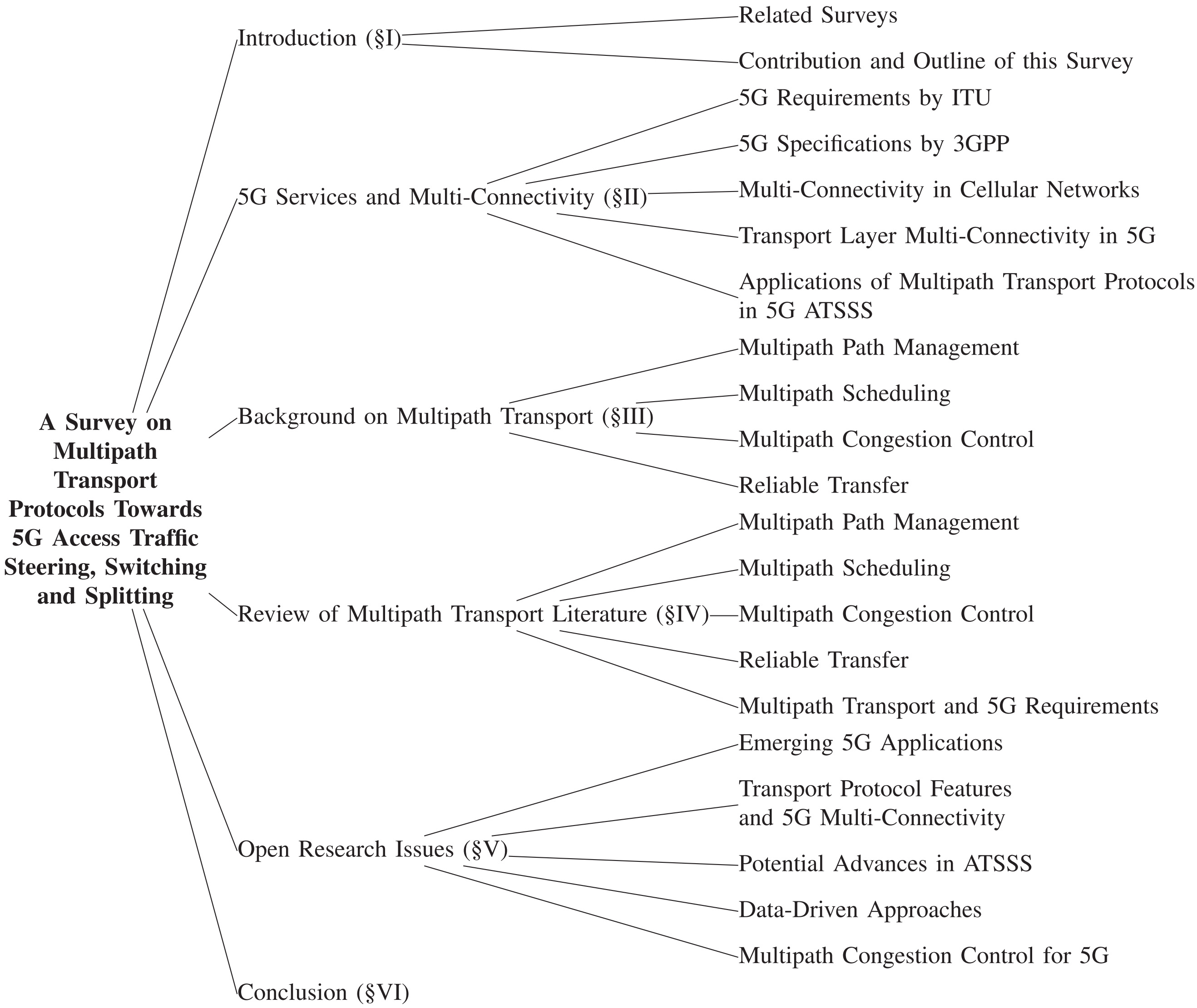


FIGURE 2. Survey structure.

and current standardization targeting 5G multi-connectivity. We then provide a bird's eye view of multipath transport protocols and their four core functionalities, i.e., path management, scheduling, congestion control, and reliable transfer in Section III. We present a comprehensive literature overview on multipath transport protocols and discuss how they fit in 5G in Section IV. Open research challenges are summarized in Section V. We conclude our work in Section VI.

II. 5G SERVICES AND MULTI-CONNECTIVITY

In this section, we present 5G requirements and specifications for eMBB and URLLC services, as first defined by ITU and then 3GPP, respectively. We then comment on the challenges for these services requirements to coexist in the network. Finally, we introduce multi-connectivity solutions in cellular systems and focus on the ATSSS architecture, which plays a key role in enabling 5G multi-connectivity and meeting eMBB and URLLC requirements. We, in particular, highlight the application of different multipath transport protocols in the standardization of ATSSS.

A. 5G REQUIREMENTS BY ITU

In early 2012, the ITU Radiocommunication sector (ITU-R) started a program to develop ''International Mobile Telecommunications (IMT) for 2020 and beyond'', preparing the stage for 5G research activities to emerge around the world. In 2015, the overall 5G requirements were settled in IMT-2020 and issued by ITU-R [2].

Therein, 5G envisages a broad variety of capabilities and applications, grouped into three main *services*, i.e., eMBB, URLLC, and mMTC, as defined in Section I. In these services, the enhanced key capabilities are captured by several parameters such as peak data rate ($Gbit/s$), latency (ms), connection density (number of devices per km^2), energy efficiency ($bit/Joule$), and spectrum efficiency ($bit/s/Hz$). More concretely, peak data rates are expected to reach 20 $Gbit/s$, which is nearly 20 times higher than IMT-Advanced (i.e., 4G systems). The energy consumption for the radio access network should be also improved by a factor at least as great as the envisaged capacity increase. Also, 5G should be able to provide 1 ms over-the-air latency to support use cases with very low latency requirements. Finally, 5G is also expected to support a connection density of up to $10^6/km^2$. A more comprehensive view of the expected enhancement of each key capability compared with IMT-Advanced is shown in Figure 3.

Further, Figure 4 shows the comparison of each key capability for eMBB, URLLC and mMTC. In eMBB, user experienced data rate, area traffic capacity, peak data rate, mobility, energy efficiency, and spectrum efficiency all have high importance. In URLLC, low latency is of the highest priority in several industrial critical applications. This key capability would be likewise required in some high mobility use cases, e.g. transportation safety. In mMTC, high connection density is needed to support a large number of devices, e.g., Internet of Things (IoT), which may intermittently use the radio access network to transmit small to large data amounts under low mobility.

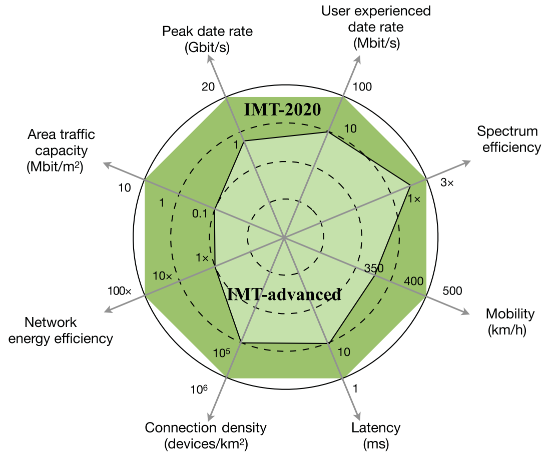


FIGURE 3. Enhancement of key capabilities from IMT-Advanced to IMT-2020.

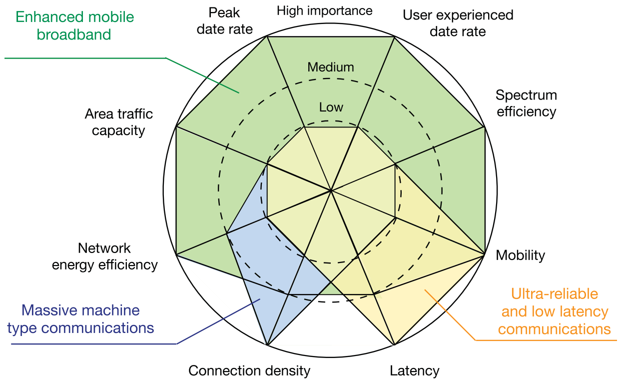


FIGURE 4. Key capability comparison among 5G services.

B. 5G SPECIFICATIONS BY 3GPP

While ITU-R sets up the general requirements of 5G, 3GPP makes the formal specifications based on those. In particular, adopting a concept referred to as *network slicing*, the 5G services proposed by ITU-R are formally mapped into a Public Land Mobile Network (PLMN) with different Slice Service Type (SST) numbers [13]. Hence, different SSTs operate within the network slicing architecture, which enables the multiplexing of virtualized and SST-dedicated logical networks on the same physical network infrastructure [17].

eMBB: 3GPP Technical Report (TR) 26.891 [18] defines eMBB as SST 1, which is suitable for handling 5G eMBB, however, not limited to consumer mobile broadband applications. As shown in [19], it is expected that SST 1 supports high data rates and high traffic density scenarios such as urban and rural wide-area (macro), indoor hotspots, dense urban, very dense crowded scenarios, high-speed trains, vehicles, and airplane connectivity.

URLLC: 3GPP TR 26.891 [18] also defines URLLC as SST 2 for use cases requiring very low latency and very high service availability, i.e. a reliability between 99.9% and 99.999%. As shown in [19], it is expected that the use cases and their respective performance requirements are derived from different industry segments and processes, e.g., industry manufacturing (industry automation), intelligent transport systems (connected cars), or electricity distribution (public critical infrastructure).

mMTC: 3GPP TR 26.891 [18] also defines mMTC as SST 3, with typical use cases being urban coverage with large cells and continuous coverage providing very high connection density of mMTC devices (massive IoT). As shown in [20], besides high connection density, mMTC also needs to maintain low power consumption to extend battery life up to 10 years.

5G Services' Coexistence: eMBB, URLLC and mMTC can also coexist as part of the same 5G network through several mechanisms, where one might be often referred to a broader term, namely, *slicing*. One of the most challenging places in the network where coexistence must be efficiently implemented is the Radio Access Network (RAN). Indeed, in the RAN, scheduling decisions are taken in order to optimally multiplex traffic from different services. For example, it is likely that RAN scheduling decisions prioritize URLLC over eMBB traffic, since URLLC cannot be queued until the next slot to wait for eMBB traffic, due to its strict latency requirements. In this direction, there are three main approaches proposed by 3GPP [21], namely, *Puncturing*, *Superposition*, and *Orthogonal* scheduler. If URLLC traffic arrives during an ongoing eMBB transmission, it can be immediately scheduled on top of eMBB, i.e., each eMBB slot is divided into mini-slots that are meant for multiplexing eMBB and URLLC traffic. Then, the gNodeB may either allocate transmission resources to both eMBB and URLLC (superposition) or temporarily interrupt eMBB traffic (puncturing) [22]. While beneficial for URLLC requirements, these methods may negatively impact the reliability of eMBB traffic. Orthogonal scheduling, on the other hand, reserves in advance (semi-static or dynamic) a number of frequency channels for URLLC. In the semi-static scheme, the gNodeB broadcasts the frame structure configuration, e.g., the current frequency numerology. In the dynamic scheme, the frame structure is frequently updated using the control channel of scheduled users, thus, incurring in higher control-plane overhead. The main drawback of this approach is to assume that URLLC traffic is always present and to reserve resources for it. Several research works also try to address such coexistence challenges. In [23], the authors propose a risk-sensitive measure to allocate resources to URLLC traffic while minimizing the risk for the eMBB traffic of achieving low rates. Hence, they propose a problem formulation that protects eMBB from drastic rate reduction while ensuring URLLC reliability. Similarly, [24] formulates an optimization problem to maximize the eMBB Minimum Expected Achieved Rate (MEAR) while provisioning URLLC, thus, focused on eMBB *puncturing*.

In all aforementioned 5G services, key enhancing capabilities related to throughput, latency or reliability may partially depend on the 5G system, e.g., radio frequency bands to achieve higher throughput, the radio protocol stack itself e.g. guaranteeing that all services can coexist. Other aspects however may be tackled by improving the interconnection and intersection between the 5G system and other infrastructure services, e.g., allowing service hosting (caching) on the 5G system from services outside the Internet (*mobile edge computing and communication*), or allowing 5G systems to leverage existing distributed cloud infrastructures for their own operation. From a different angle, how UEs connect and use the 5G system can be further leveraged. Therefore, in this paper we focus on this latter aspect, pointing to how the proven benefits of multi-connectivity and, more specifically, of multipath transport [25]–[27], can aid 5G services to reach their key performance indicators. In more detail, we focus on aspects such as high throughput in eMBB, and low latency and high reliability in URLLC.

C. MULTI-CONNECTIVITY IN CELLULAR NETWORKS

Multi-connectivity is one of the paradigms in 5G that aims to satisfy the service requirements defined in Section II-A. We provide in this section a brief overview on existing multi-connectivity solutions for cellular networks, along with related standardization activities.

One of the first multi-connectivity solutions was introduced in 3GPP Rel-8 (2008) and referred to as Access Network Discovery and Selection Function (ANDSF) [28]. In particular, ANDSF targets interoperability between 3GPP and non-3GPP systems. Focusing on the cellular access, Coordinated Multi-Point (CoMP) was introduced in Rel-11 (2012). In this case, multiple base stations can transmit (receive) in parallel the same data towards a UE, in order to improve the communication quality in poor coverage areas. While CoMP lies across physical and MAC layers, Dual Connectivity (DC) is performed in the above Packet Data Convergence Protocol (PDCP) layer. Standardized in Rel-12 (2015), DC allows a UE to exploit two not co-located LTE access nodes, e.g., two evolved Node Bs (eNBs). The Master eNB terminates the control plane in the LTE core and coordinates with the Secondary eNB to provide additional radio resources to the UE.

Similar solutions are then introduced for non-3GPP access in Rel-13 (2016) and extended in Rel-14 (2017). They are referred to as LTE-WLAN Aggregation (LWA) and LTE-WLAN radio-level integration with IP security tunnel (LWIP). In both cases, the WiFi access point has a similar scope compared to a Secondary eNB in DC, and can be co-located or not with the primary access node. The user device reports WiFi-related measurements to the cellular network, which decides to activate or not the multi-connectivity option. The WiFi traffic is managed within the LTE system via specific adaptation protocols [29]. Mechanisms similar to LWA and LWIP can be envisioned for 5G [30]. However, initial proposals in Rel-15 (2018) were focused on cellular access, and have led to extending DC to support parallel use of LTE and 5G NR.

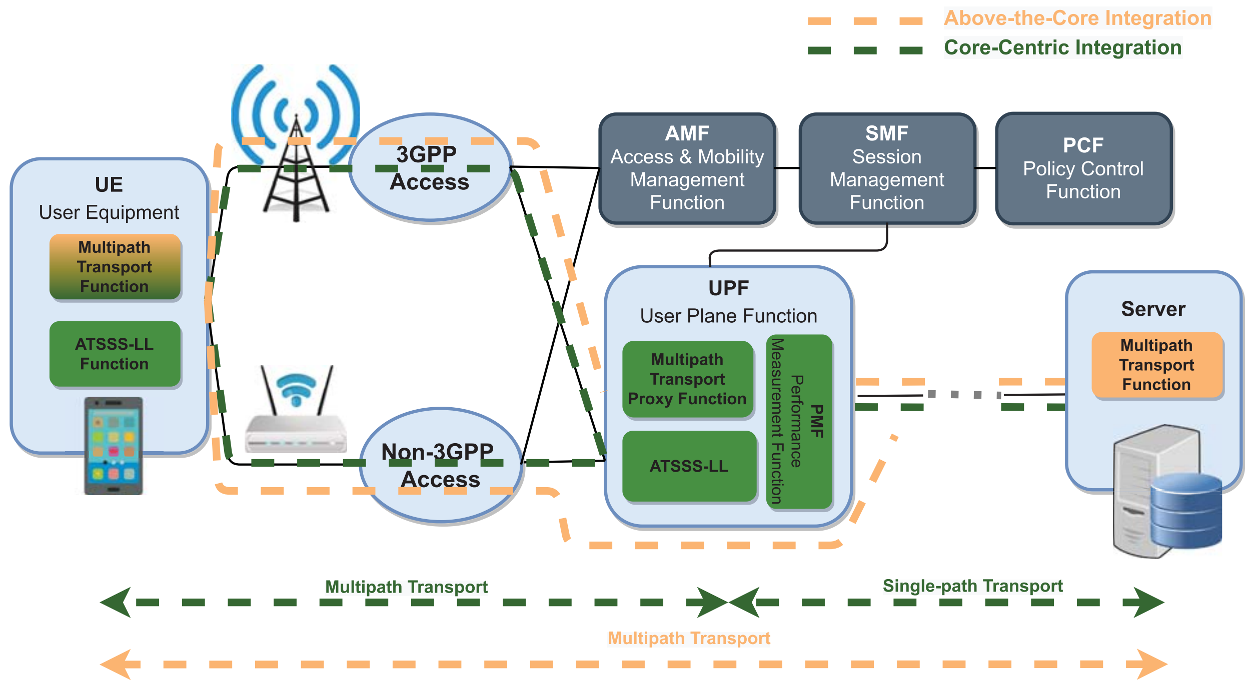


FIGURE 5. High-level view of above-the-core and core-centric integration options in 5G. For the second option, the main 5G functional blocks involved in ATSSS architecture are reported.

In the next section, we focus on two main approaches that aim at integrating multipath transport solutions to enable multi-connectivity in 5G systems. After introducing two main options, we detail the ATSSS architecture, which is one of the main multi-connectivity frameworks for 5G. Proposed in Rel-16, ATSSS proposes a direct integration and use of multipath transport protocols in 5G systems.

D. TRANSPORT LAYER MULTI-CONNECTIVITY IN 5G

Currently, two main approaches are highlighted to tackle 5G multi-connectivity via multipath transport solutions: *Above-the-Core* and *Core-Centric*. In the Above-the-Core integration, the multipath transport protocol is deployed at the client and the server sides, and the aggregation of different paths occurs in between, without impacting the network. In the Core-Centric integration, the multipath transport protocol is deployed at the client and in the 5G Core (i.e., through a multipath proxy), and single path transport is run between the core network and the server. A high-level view of both approaches is shown in Figure 5, with yellow and green dashed lines representing Above-the-Core and Core-Centric, respectively.

The Above-the-Core integration has been prevalent in academia and industry, with several contributions. For example, several early efforts show the benefits of multipath transport protocols in smartphones [31]–[35], which could be seen as predecessors of the ongoing standardization activities in 5G multi-connectivity. The goal of the first measurement studies was to evaluate whether the proven benefits of multipath transport in data center networks could be also leveraged by multi-homed devices, e.g., smartphones, and by network operators, e.g., to offload cellular network traffic to WLAN. The majority of these contributions use Multipath Transmission Control Protocol (MPTCP) as the base transport protocol, with exception of [36], that presents both MPTCP and Multipath QUIC (MPQUIC). Few others focus on developing tools to tune application and transport protocol interaction to improve performance and battery life [37], [38]. It is consistently demonstrated that multipath transport can mitigate the impact of handover in applications under mobility, e.g., when moving between WLAN and cellular coverage. Since early experiments in 2013, this aspect has been particularly supported by iPhone devices [39], and also more recently in 2020 by Alibaba and Apple and [40].

The Core-Centric integration, as highlighted by several use cases [41], [42], is a stronger candidate to be adopted in 5G systems, since it enables a more direct control of multi-connectivity within the cellular system.

3GPP has specified the ATSSS architecture in TS 23.501 Rel-16, as an instantiation of the Core-Centric approach. The key concept being introduced is the Multi-Access Protocol Data Unit (MA PDU) session. The MA PDU session generalizes the single-access PDU session and allows an application to send/receive traffic over 3GPP access, non-3GPP access, or both simultaneously. The MA PDU session is enabled in the ATSSS architecture, which is depicted in Figure 5; it is established between the User Equipment (UE) and User Plane Function (UPF), with both 3GPP and non-3GPP access networks in the middle. Moreover, as shown in Figure 5, other 5G core network functions are involved in the ATSSS operation, i.e., Access and Mobility Management Function (AMF), Session Management Function (SMF), and Policy Control function (PCF). Once a MA PDU session is established, it handles the traffic over different networks via Steering, Switching, and Splitting functions, defined as follows:

- *Steering:* It enables the selection and use of an access network for a data flow;
- *Switching:* It allows to redirect all traffic of an ongoing data flow from one access network to another, while maintaining service continuity;
- *Splitting:* It enables the splitting of the traffic of a data flow across multiple access networks, so that some traffic of the data flow is transferred via one access and some other traffic of the same data flow is transferred via another access.

As shown in Figure 5, the PCF controls ATSSS by delivering the policy rule to the SMF. The policy rule, shared by the SMF with the UE (uplink) or the UPF (downlink), contains the indication on which ATSSS *steering function* and *steering mode* to adopt. To simplify the terminology from TS 23.501, we refer in the following to only steering, when referring to Steering, Switching, or Splitting.

With the notion of MA PDU introduced by ATSSS, there are several options for fine grained control of data flows to be served over one or more access networks. For example, Steering selects, across several available access networks, the one that better fulfills a certain mode, e.g., smallest delay, etc. Switching, on the other hand, takes a hard decision to abandon one of the access networks and invariably use either one access network or another, e.g., enabling connection migration and handover mechanisms. Splitting allows for using (two or more) access networks simultaneously, transferring different parts of a data flow on each available access network. Finally, Splitting allows for selecting a particular access network to provide, e.g., redundancy, or both access networks to provide, e.g, aggregation. As further detailed below, multipath transport protocols plays a key role to realize such functionality.

E. APPLICATIONS OF MULTIPATH TRANSPORT PROTOCOLS IN 5G ATSSS

TS 23.501 defines two ways of implementing steering functionalities: a) the use of a multipath transport protocol, above the IP layer, and b) the use of a so-called ATSSS Lower Layer (ATSSS-LL), below the IP layer. In the case of multipath transport, as shown in Figure 5, the UE and UPF communicate through the Multipath Transport Function (in the UE) and the Multipath Transport Proxy Function (in the UPF). In the case of ATSSS-LL, the UE and UPF communicate with each other via the combination of ATSSS-LL Function of the UE and UPF. In addition, UPF supports Performance Measurement Functionality (PMF), that may be used by the MA PDU session to obtain access performance measurements over 3GPP and/or non-3GPP access networks.

While there is no recommendation on the method for ATSSS-LL yet in Rel-16, TS 23.501 Rel-16 identifies a specific multipath transport protocol for the multipath transport functionality, i.e., MPTCP. However, in the studies that lead up to 3GPP Rel-16 more protocols were analyzed for ATSSS support in the 5G System architecture. During these studies, recorded in TR 23.793 [15], the use of QUIC, MPQUIC, Stream Control Transmission Protocol (SCTP), and multipath User Datagram Protocol (UDP) were considered.

In terms of steering modes, TS 23.501 defines four different modes that can be used with ATSSS, as follows:

- *Active-Standby:* The traffic of an MA-PDU session is sent to one access network only, referred to as ''active'' access. The other access network is in ''standby'' and takes traffic only when the active one is unavailable. The active access is defined when the MA-PDU session is established and can remain the same or change during the session lifetime;
- *Priority-based:* Some priority weights are assigned to the available access networks either statically during the establishment of a MA-PDU session or dynamically during the lifetime of a MA-PDU session. The traffic is managed by the higher priority access; however, when it is congested or unavailable, the traffic is redirected onto the lower priority access;
- *Smallest Delay:* The used access network is the one providing the shortest Round Trip Time (RTT). It conceptually belongs to the Priority-based mode but, in this case, the higher priority access is determined dynamically in the lifetime of an MA-PDU session, based on RTT measurements;
- *Load-balancing:* Each access network receives a percentage of the data of the MA-PDU session, depending on the assigned weight factor. If one access becomes unavailable, all traffic is sent to the other.

Moreover, two further modes were under discussion in TR 23.793, and can be foreseen as possible extensions for future ATSSS specifications:

- *Best-Access:* It generalizes the Smallest Delay mode, making it possible to adopt other factors rather than RTT to decide the access network with the best performance to use. It also conceptually belongs to the Priority-based mode, but in this case, the higher priority access is also determined dynamically during the lifetime of an MA-PDU session, based on the performance of the access;
- *Redundant:* All or some data flows are transmitted on both accesses in order to increase reliability.

The above steering modes are all supported by MPTCP.

The standardization of 3GPP Rel-17 and ATSSS Phase 2 is at the time of writing of this article on-going work, expected to conclude in March 2022. Phase 2 is focused on defining several improvements of the steering modes, such as different ways of controlling the load balancing or determining when a path is congested for the priority-based steering mode. However, the studies preceding ATSSS Phase 2 [43] again considered additional steering functionalities based on both QUIC and MP-QUIC, as well as adding a QUIC-based proxy (with and without multipath capability). The latter still depends on work to be carried out at the IETF.

Further, the ATSSS Phase 2 study item hints to an ATSSS Phase 3, including features and scenarios that are out-of-scope in ATSSS Phase 2, e.g., a MA PDU session with more than two network paths. Discussions on what study items to include for Rel-18 are ongoing at the time of writing. QUIC and MP-QUIC are again under discussion and Multipath DCCP [44] has also been suggested as an option. As Rel-17 is still ongoing work and standardization of Rel-18 has not yet started, we opt to base the work in this survey on ATSSS Release 16 [13], [15], which already lay the foundation for the on-going work in the subsequent ATSSS phases.

The key role of multipath transport protocols in ATSSS motivates us to investigate multipath transport protocols more generally in the literature. We report our review and analysis in Sections III and IV, respectively, where we also highlight a) the connection to the ATSSS architecture and mapping with ATSSS steering modes considered for ATSSS Phase 1, and b) how the proposed multipath schemes may help towards satisfying the requirements of 5G eMBB and URLLC services.

III. BACKGROUND ON MULTIPATH TRANSPORT

Multipath transport protocols are designed to improve both communication throughput and resilience as they are able to leverage several network paths simultaneously and seamlessly support failover. We note that all three transport protocols considered in this survey, namely, SCTP, TCP and QUIC, have different multipath features supporting at least one of the multipath benefits (throughput and/or resilience). For example, SCTP is already able to leverage multiple paths, however, it uses one primary path while others are meant for failover, when the primary path fails. Thus, SCTP is able to natively improve resilience. Also, QUIC with its *connection migration* feature is able to move a connection across network accesses, thus, also improving failover. TCP, on the other hand, does not natively support failover as it ties IP addresses and ports to identify connections. None of the single-path implementations are able to improve throughput, which is, beyond improved resilience, one of the main promised benefits of their multipath counterparts, which we refer to in more details in the following. The realisation of the multipath connection depends on the protocol implementation specifics. Figure 6 depicts a high-level representation of the single path (left-hand side) and the multipath (right-hand side) protocol stacks. Nowadays, three main multipath protocols exist, i.e., Concurrent Multipath Transfer SCTP (CMT-SCTP), MPTCP, and MPQUIC, which are the focuses of this survey. In addition, IETF recently has also initiated work on extending the Datagram Congestion Control Protocol (DCCP) [45] to support the multipath operation, aiming to deliver Multipath DCCP (MP-DCCP) [44].

As an extension of SCTP, CMT-SCTP [25], [46] is one of the first multipath transport protocols that considered the

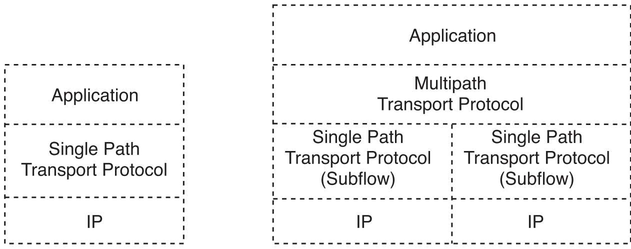


FIGURE 6. Single path and multipath transport protocol stack representations.

simultaneous data transfer over different paths. MPTCP [47] implements the multipath extension of the most widely used transport layer protocol, TCP. It is designed to be transparent to both higher and lower layers, in order to counteract the proliferation of middleboxes in the Internet that hinder the deployment of new transport protocols [48]. Recently, IETF QUIC[2] became an attractive alternative to TCP since it can combine the benefits of HTTP/2, Transport Layer Security (TLS) and TCP over UDP to reduce latency and improve security. QUIC encrypts all payload and most of the protocol headers to prevent interference from middleboxes [50]. Motivated by the success of MPTCP and the interest in QUIC by both industry and academia [51], [52], the multipath extension for QUIC (MPQUIC) is proposed in [53], [54], with similarities to MPTCP.

Note that, MPTCP plays a central role in ATSSS while MPQUIC has been discussed as an alternative. In this paper, as much as we would like to keep the discussions more general on multipath transport protocols, due to its maturity and adoption in the community we refer more often to MPTCP literature.

Despite different transport protocol design and implementations, all above mentioned multipath transport protocols share four common functionalities, which are of relevance in ATSSS:

- The *multipath path management*, which is in charge of initiating and managing the connections, i.e., subflows, part of the same multipath connection.
- The *multipath scheduling*, which is in charge of distributing packets over different paths following a certain policy, e.g., aggregated throughput (utilise all available capacity), reduce latency (prefer low latency paths) or improve reliability (duplicate packets).
- The *multipath congestion control*, which aims to detect network congestion, adjust the sender rate accordingly (as in the single path case), and deal with other aspects of a multipath transmission, e.g., fairness towards single path traffic.

[2]The Google QUIC protocol (gQUIC) is the original implementation [49] adopted by the IETF for standardization. However, gQUIC and IETF QUIC are today two different implementations, where IETF QUIC significantly diverges from the original gQUIC proposal in terms of the handshake, wire format of the packets, or adaptation to Hypertext Transfer Protocol (HTTP), among other major differences.

- The *reliable transfer*, which is in charge of loss detection and loss recovery (as in the single path case) by having a mechanism at the sender that detects packet losses and an associate mechanism in charge of recovering these packets with retransmissions.

Next, we describe these main functionalities. In Section IV, we will review the state-of-the-art literature for these functionalities and provide a direct mapping of them to the ATSSS modes.

A. MULTIPATH PATH MANAGEMENT

The path manager component determines what path to use for connection establishment and when and how additional subflows are established, and it can also control the advertisement or acceptance of available IP addresses for new subflows. This logic generally depends on the application requirements, e.g., some applications use multipath only for handover while others use it for load sharing. In general, however, the combination of how and when subflows are established with how the subflows are used during the connection, e.g., how packets are distributed over them, is performed in conjunction with the multipath scheduler, described in next section. For instance, the path management algorithm can establish a subflow over each of two paths, and the scheduler, e.g., by means of measuring the RTT of the subflows, can prefer the subflow with the lowest RTT. This operation mode describes very closely the default path management and scheduling operations in MPTCP. To better understand how a path manager operates in MPTCP, we provide an example, illustrated in Figure 7: Host A signals to Host B the support for MPTCP via a `MP_CAPABLE` TCP option during the initial handshake. Once the initial subflow is established, the `MP_JOIN` option is sent to associate a new subflow to the existing MPTCP connection. If Host A gets a new IP address during the connection, `MP_ADD` is signalled by MPTCP, telling Host B about the new address, where a new subflow can be established. For example, if Host A and Host B have initially two IP addresses each, and all possible subflows are established, the multipath connection results in a *full-mesh* of subflows, i.e., A1-B1, A1-B2, A2-B1, A2-B2. If Host A gets a new address, denoted A3, during the connection, it can signal this address to Host B, and additional subflows can be added to the multipath connection, i.e., A3-B1 and A3-B2.

In MPTCP, there are currently three implementations for path management:

- *Default* neither announces IP addresses nor initiates the creation of new subflows, as it only accepts their passive creation, e.g., a request from the remote host;
- *Fullmesh* establishes the full-mesh of subflows according to the available IP addresses, similar to the previous example with Host A and Host B (see Figure 7);
- *Ndiffports* uses the same pair of IP addresses, where each subflow has a different source TCP port.

The path management in MPQUIC is specified with a different approach [55], [56]: during the handshake, both hosts

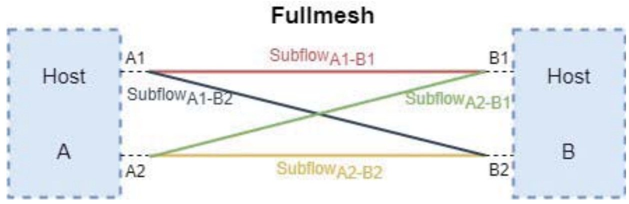


FIGURE 7. Illustration of the fullmesh path management algorithm.

can negotiate the multipath capability via frames, and path management can be implemented via the `PATH_STATUS` frame. With this frame, the hosts can signal preference or claim the state for a subflow, e.g., set the subflow as available, standby, mark its priority or simply abandon it. To validate a path, i.e., probe it, `PATH_CHALLENGE` and `PATH_RESPONSE` frames can be used. As regards SCTP, during the association startup, a primary path is defined for each SCTP host and used for sending SCTP packets, where all other paths are used for failover or used for retransmissions. The IP addresses in a SCTP association are exchanged and verified during association setup, and each destination address is a different path towards the corresponding host. The path reachability is verified with heartbeat chunks sent periodically to all destinations. Dynamic Address Reconfiguration (DAR) is an SCTP extension for SCTP's multihoming, and enables to dynamically add or delete IP addresses, and to request a primary-path change during an active SCTP association.

B. MULTIPATH SCHEDULING

The multipath scheduler component is primarily in charge of distributing data over available paths according to the given policy. The available paths can be classified as homogeneous or heterogeneous, depending on how similar they are in terms of bandwidth, delay, loss rates, and other characteristics [57]. Prior to using the paths, they need to be established at the beginning or during the multipath connection by the path manager, see Section III-A.

To illustrate the challenges involved in scheduling, let us consider a basic Round-Robin (RR) scheduler. In MPTCP, RR cyclically sends packets over each path, as long as there is space in their Congestion Windows (CWND). While this is a very simple approach that may work reasonably for homogeneous paths, RR is not very useful in practice as it does not account for path heterogeneity. Since RR does not use any characteristics of the paths in the scheduling decision, the packets may arrive out-of-order, which causes receiver buffer blocking and head-of-line blocking when data can be only delivered to the application in-order, thus, decreasing overall performance [58], [59]. In general, as path heterogeneity increases, scheduling and making use of multiple paths gets more challenging.

There are different ways to tackle multipath scheduling performance challenges. For example, the scheduler can use transport layer information, e.g., RTT and CWND, to estimate the transfer time of each packet on each path. Based on the estimation, the scheduler tries to distribute packets so that they arrive in order [60]–[62]. Alternatively, the scheduler can duplicate packets to provide low latency or high reliability. The need depends on the current path status and the optimization goal (throughput or latency). More recently, machine learning approaches (e.g., reinforcement or supervised learning, etc.) are used as ways to enable latency and/or throughput optimization in the same algorithm. Here, machine learning features can be derived from transport layer information such as RTT, CWND, inflight packets [63]–[65], etc.

C. MULTIPATH CONGESTION CONTROL

Traditionally designed for single-path TCP scenarios, congestion control algorithms operate on packet-level characteristics such as loss and delay to detect network congestion and react accordingly, e.g., by adjusting the sending rate. Among other requirements, there is a fairness notion that guarantees the same resources for each TCP flow, e.g., the same bandwidth at the shared bottleneck [66].

However, the emergence of multipath transport protocols brought the need to revisit the fairness aspect. In the case of CMT-SCTP, the protocol treats all paths belonging to a multipath connection separately, applying single-path congestion control over each path independently. In MPTCP, the fairness aspect is part of its three design goals, as discussed in [66]–[68]:

1) *Improve Throughput:* A multipath flow should perform at least as well as a single path flow would on the best available path;
2) *Do Not Harm:* On each path, a multipath flow should not take more resources than other single path flows;
3) *Balance Congestion:* A multipath flow should move as much traffic as possible off its most congested paths, subject to meeting the first two goals.

Requirement 2) has driven specifically the design of several algorithms, and it is mentioned as ‘‘fairness in the broader, network sense’’ in RFC6356 [69]. When it comes to MPQUIC, it is still unclear which direction standardization will take. Initial research-oriented proposals [27], [70] suggest a design similar to MPTCP. More generally, multipath congestion control is categorised into *uncoupled* and *coupled* approaches. The uncoupled proposals treat each of the subflows of a single multipath connection as individual connections, i.e., their CWND is increased or reduced without considering other subflows. However, for the sake of standardization, the coupled proposals were adopted, as described in RFC6356, since it treats all subflows belonging to the multipath connection as a single connection. In MPTCP, the increase of all CWNDs of the subflows from the same multipath connection should not exceed that of a single TCP connection, thus not unfairly interacting with single path traffic. The CWND decrease, however, is handled individually, since if one of the paths is more congested than others, the subflow of the multipath connection should back-off as single-path traffic would do.

D. RELIABLE TRANSFER

Transport layer protocols are mainly distinguished by providing reliable or unreliable data transfer. For instance, all data sent over TCP is guaranteed to be delivered, i.e, TCP is fully-reliable. UDP, on the other hand, does not keep track of lost or corrupted packets, i.e., it does not provide guaranteed delivery and it is unreliable.[3] SCTP on the other hand also implements partial reliability, i.e., some level of packet loss can be tolerated. Multipath variants of these protocols implement reliability as in their single-path counterparts, however, with functionalities specifically meant for multipath. For example, in MPTCP, as long as packet loss is recovered by a fast retransmit, i.e., the receiver sends Duplicated ACKS (DupACKs) to signal missing packet(s) to the sender; these packets are recovered in the same subflow. Otherwise, if packet loss is detected by a Retransmission Timeout (RTO), they are also retransmitted on other subflow(s).

These loss detection and recovery mechanisms were designed with some assumptions about the underlying networks and they are known to perform suboptimally in some cases, especially when delay and loss rates are high [72]. Therefore, there is interest to apply approaches such as Forward Error Correction (FEC) and Network Coding (NC) [73] in transport protocols. In FEC, input data is encoded at the sender resulting in a combination of source and repair packets, where repair packets are used to recover lost packets at the receiver. On the other hand, NC can be performed at the sender and on intermediate nodes (all or a subset of them).

In the past, different FEC and NC algorithms have been proposed inside the transport layer, in particular for TCP, where the implementations were often in conflict with the congestion control operation and prohibitively complex [74]. For multipath, FEC and NC mechanisms are applied in the subflow level [75], [76], i.e., in the single path transport protocol connection (subflow) to alleviate the heterogeneity of the underlying paths, especially when these have different loss rates.

IV. REVIEW OF MULTIPATH TRANSPORT LITERATURE

This section provides a review of literature addressing main aspects of the functionalities of multipath transport protocols, i.e., path management, scheduling, congestion control, and reliable transfer, as introduced in Section III. In particular, the reviewed works are categorized in terms of ATSSS modes, in order to emphasize how they can be exploited in ATSSS to improve 5G multi-connectivity, ultimately contributing to achieving eMBB and URLLC service requirements.

A. MULTIPATH PATH MANAGEMENT

Besides path establishment provisioning, the path manager can support the implementation of the ATSSS modes. More specifically, the implementation of handover closely resembles the Active-Standby ATSSS steering mode, where a path

[3]In QUIC, even though it is implemented on top of UDP, the reliability mechanisms are present and designed following the ideas in TCP [71].

manager tries to use the active network access and switch to the standby path after a certain number of retransmissions. This operation mode is very similar to MPTCP in Apple iPhones. Similarly, a path manager that establishes subflows over all paths combined with a packet scheduler that favors subflows with the lowest RTT resembles Smallest Delay ATSSS steering mode as well as MPTCP's Linux operation.

Exploring path management in handover scenarios, [33] performs a real-world study using WiFi/3G to show that MPTCP maintains application connectivity when moving between network connections. In addition, by sparing one bit as an *echo* bit in *Remove Address* and *Add Address* options, and [33] develops a scheme to tackle application degradation. Depending on the active states of paths during handover, [33] shows that MPTCP can decrease the application delay for VoIP up to 20 times than single path TCP. In [77] the authors summarize the points that should be considered on the radio access network when implementing data offloading with MPTCP covering complementary coverage, spectrum aggregation and utilization, radio planning, RF load balancing, channel holding time, deployment, backhaul capacity, and mobility. Reference [78] investigates handover connection disruption and glitches with MPTCP. To improve service continuity during handovers, it uses a proactive cross-layer assisted mechanism with Signal-To-Noise (SNR) and Bandwidth-Delay Product (BDP) based CWND adjustments. During handover, mechanism proposed in [78] can reach 2 to 5 times throughput increase compared with the default MPTCP. Reference [79] presents an analytical model for multipath WiFi/cellular handover, which derives the aggregate handover time, providing a tool for tuning the cellular bitrate to satisfy the users' transmission requirements while maximizing the network resource utilization efficiency.

In the path management implementations available in MPTCP, *full-mesh* might be more suitable for Internet scenarios, exploring all possible combinations between IP addresses of two hosts, thus, supporting applications that aim at load balancing or improving throughput. Similarly, *Ndiffports* was originally designed for datacenter networks to enable load-balanced paths with Equal Cost Multipath (ECMP) [80]. *Default* is implemented to passively accept the creation of new subflows. Finally, while not a path manager in the strict sense, *binder* [81] focuses on community networks to help applications to benefit from gateway aggregation using loose source routing. For the examined scenario of dramatic network changes in the period of 3 seconds, *binder* consistently outperforms TCP baseline with an improvement ranging from 20% to 60%.

In MPQUIC the design of path management is on-going work, where, so far, it is taking a different approach compared to MPTCP. To date, in MPQUIC, the proposal is that hosts can signal and negotiate via frames how to establish and use network paths during the connection lifetime, see Section III-A. In other words there are no path management implementations serving different scenarios such as it is the case in MPTCP.

Finally, leveraging SCTP for path handover, [82]–[85] and several others surveyed in [86] cite the problems of spurious retransmissions, unnecessary CWND reductions and reordering caused by path handover. Reference [87] evaluates the feasibility to combine both CMT and SCTP with dynamic address reconfiguration as a potential enhancement to the handover schemes.

B. MULTIPATH SCHEDULING

Multipath scheduling is in charge of distributing data onto different paths. This is a core function in a multipath transport protocol, since wrong scheduling decisions can lead to performance decrease, particularly due to out-of-order data delivery at the receiver. In [25], [59], it is shown that Round-Robin (RR) is only effective when paths are homogeneous. Thus, to guarantee multipath transport performance enhancements with any combination of network paths, many scheduling approaches have been proposed. We categorize them into six categories resembling ATSSS steering modes introduced in Section II-E: 1) Smallest Delay, 2) Best-Access, 3) Priority-based, 4) Load-balancing, 5) Redundant, and 6) Active-Standby. Note that, while many multipath scheduler algorithms are applicable accross protocols, some algorithms may explore features of the transport protocols that may be available in one implementation but not in the other, e.g., the notion of *streams* in CMT-SCTP and MPQUIC is absent in MPTCP. In the following, we refer to stream-based multipath schedulers when applicable to each of the ATSSS categories.

We notice that minRTT is by definition the only scheduling algorithm mapping to Smallest Delay steering mode. It is the default algorithm in MPTCP [88], and prioritizes the path with the lowest estimated RTT, if it is not congested. As the congestion level increases, minRTT redirects traffic on the other paths.

In Best-Access steering mode, the definition of ''best'' is generally referred to as the best performance, but refers to the estimated latency of the paths in most of the literature. The estimated latency uses other features besides RTT as information, thus somehow extending the Smallest Delay steering mode. Out-of-order Transfer for In-order Arrival (OTIAS) [89] is, in many aspects, similar to Earliest Completion First (ECF) [61]. They apply a proactive approach [62], i.e., the path with the shortest transfer time is prioritized regardless of the congestion level (CWND space). Therefore, the transfer time also includes the time of waiting for the space in the CWND, which is different from a reactive approach, e.g., minRTT. The work in [90] applies a similar approach but taking stream priority in MPQUIC into account. Blocking Estimation-based (BLEST) [60] reduces receiver buffer blocking by prioritizing the faster path using RTT, inflight packets, CWND, and send window size as estimates. Shortest Transmission Time First (STTF) [62] comes as a fine-grained shortest delay version of BLEST targeting short flows. Compared with the default scheduler minRTT, BLEST and STTF are particularly shown to reduce web object transmission times with up to 51% and provide 45% faster communication for interactive applications. In terms of cross-layer approaches, Quality Aware (QAware) [91] incorporates the local queue buffer occupancy information of the Network Interface Card (NIC), aiming to improve the estimation of end-to-end delay. QAware can provide an improvement with up to 37% over the minRTT scheduler. The work in [92] focuses on throughput in cloud networking, creating subflows over disjoint paths, and using cross-layer information from MPTCP and the Location Identifier Separation Protocol (LISP) to learn about paths' diversity. Furthermore, [93] proposes a cross-layer scheduler for video streaming, using both application and transport layer information. Different from the above approaches, [94] proposes the client-based multipath TCP (cMPTCP) that aims to be deployed over multiple LTE networks from different operators. cMPTCP utilizes the client to infer the bottleneck state of an end-to-end MPTCP connection, aiming for a more accurate selection of the best-access. cMPTCP is shown to outperform the default minRTT with up to 18.5% and the other state-of-the-art multipath schedulers such as ECF with up to 11.7% for the download throughput. Applying machine learning to determine the best-access path leads to another set of approaches. Reles [63] uses offline reinforcement learning, i.e., a Deep Q-Network (DQN), to train a multipath scheduler with throughput as the reward, and delay and packet loss as penalties. A similar approach is applied in [64], where a penalty is given when the number of unacknowledged packets exceeds a limit. Applying online learning in MPQUIC, Peekaboo [65] proposes a multipath scheduler that is aware of the dynamics of the paths and can adapt its scheduling strategy accordingly. More recently, [95] proposes an enhancement to Peekaboo, i.e., M-Peekaboo, capable of handling high oscillations in terms of network path characteristics, i.e, delay, bandwidth and loss, observed in 5G millimeter wave network paths. These scheduling approaches based on machine learning in general outperform the selected scheduling approaches that are not based on machine learning. For example, M-Peekaboo is shown to outperform BLEST with up to 28.7% in the emulated 5G networks.

The Priority-based steering mode can include both the Smallest Delay mode and Best-Access mode. However, we try to assign the works from literature to the steering mode that is as specific as possible, thus only covering the works that are exclusive in the Priority-based steering mode here. The works belonging to this steering mode use pre-defined priority information to influence the scheduling decision. MP-DASH [96] proposes a scheduling framework for video streaming that is aware of network interface preferences from users, e.g., prioritizing WiFi over cellular links. The scheduling decision is deducted by solving an integer programming problem to minimize the usage of the unwanted path while trying to meet users' Quality of Experience (QoE) requirements. The results indicate that MP-DASH can reduce cellular usage by up to 99% and radio energy consumption by up to 85% with negligible degradation of QoE, compared with off-the-shelf MPTCP. The work in [97] adopts the purchased price of the path as the prior information. It is assumed that, under a guaranteed throughput, the users prefer to use the path having lower costs. Then, by applying Lyapunov optimization, the paper aims to maximize the throughput while minimizing the price cost for users. Also adopting the path cost to derive the priority, [98] proposes a cost-based scheduling algorithm, which simultaneously reduces the cost of multipath use for network operators and also retains the QoE levels required by the end-users in case of bursty video-on-demand traffic. Both [97] and [98] present that it is possible to maintain the performance metric as the default minRTT, while decreasing the cost ranging from 20% to more than 50%.

The goal of Load-balancing steering mode is to assign a number of packets for each path, aiming to balance the load over different paths. The main difference with the other steering modes is that it directly schedules a group of packets together, while the other steering modes schedule on a per-packet basis. When assigning packets to paths, the existing works usually take into account the capacity and latency of each path. Forward Prediction Scheduling (FPS) [99] predicts the packets' arrival time and sends packets in a manner that they are expected to be received. Delay Aware Packet Scheduling (DAPS) [100] assigns the number of packets over each path based on the ratio of RTT between the paths. Reference [101] considers the RTT of different paths for load-balancing at the sender side to specifically rearrange the transmission order of the packets. An approach for actively sensing the paths' status and quality is proposed in [102], aiming to address the inaccurate estimation of the path latency when assigning the load to each path, caused by the underlying wireless networks. References [99], [100], [101], and [102] mainly provide comparisons with RR and show throughput increase ranging from 10% to 40%. The mechanism proposed in [103] is tailored for lossy networks and takes loss rates into account to model and estimate latency and data amount to send on each path. According to the experimental results, [103] can increase the download throughput around 10% compared to FPS. Reference [104] argues that although pre-allocating packets over different paths seem to ensure in-order-arrival, there often exists a mismatch between the estimated and the real transfer time, especially in wireless networks. To compensate for the inaccurate estimation, a gap composed of several packets that are not yet scheduled is left between the packets sent over different paths, and is self-adjusted based on ACKs which can reflect the out-of-order arrival degree. Reference [104] shows throughput improvements of up to 30% when the in-network buffer is limited, and 15% when the host buffer is limited, compared to ECF. In [105], PStream explores priority-based stream in MPQUIC making use of stream to alleviate head-of-line-blocking in heterogeneous environments. Evaluation shows that PStream can reduce up to 25.4% of page load time in high path heterogeneity, compared to minRTT. Focusing on video

QoE [106], NineTails is a multipath MPQUIC scheduler that utilizes selective multipath redundancy to control tail loss and near-tail loss latencies in heterogeneous wireless networks. With this design thought, NineTails is shown to decrease the tail application latency up to 18%.

In Redundant steering mode, some or all packets are duplicated to enhance the service reliability and obtain the lowest latency over different paths. ReMP [107] proposes duplication for all the packets to reduce latency and increase reliability. For a real world mobile scenario in a stressed dynamic environment, ReMP TCP can halve the average round-trip time and reduce its standard deviation by a factor of 19. However, this comes with a substantial bandwidth overhead. Hence, several multipath scheduling algorithms categorized under Redundant mode actually provide advanced solutions, which in essence combine Redundant with other ATSSS steering modes, ultimately paving the way towards ATSSS enhancements. For example, leveraging selective packet duplication, the work in [108] proposes an adaptive mechanism that duplicates packets only when a path degrades, estimated by observing one-way-delay fluctuations, and combines this with the Load-balancing mode. Targeting vehicle-type applications, Redundancy-Aided VEhicular Networking (RAVEN) [109] proposes a trade-off between data usage and duplication degree, introducing a confidence interval in the scheduling: If all packets are duplicated, 100% confidence interval is achieved. In its mechanisms, RAVEN jointly covers Redundant and Smallest delay steering modes. Nevertheless, although significant gains ranging from 24.5% to 53.2% are obtained, both [108] and [109] only provide comparisons against default minRTT. [110] proposes REdundant Diversity scheduling (RED) which prioritizes packet replication by uncorrelated paths, selecting paths and replicates packets based on the Spearman's correlation coefficient. Compared with pure redundant multipath scheduler, like ReMP, RED can achieve up to 30% higher throughput. Similarly, but targeting high loss networks, [111] proposes an adaptive duplication scheme based on the estimation of the per path loss rate, thus balancing Redundant and Best-access modes, the latter using RTT and loss rate in order to determine the path latency. While the proposed approach maintains mean delay at the same level of ReMP and up to 3 times smaller than minRTT, it can reach up to 2 times of throughput over ReMP and at the same level of minRTT.

The Active-Standby steering mode utilizes only one active path for transmission while other paths are used for backup. Thus, it focuses on the seamless handover applied in multipath, which is also studied in the literature [33], [36], [82]. Furthermore, several multipath proposals lie across multiple ATSSS steering modes. For example, the mechanism introduced in [58] initially tries all paths and suspend the path with a low score to ensure in-order packet delivery, ultimately combining Active-Standby with Smallest delay. Similarly, [112] decides if the scheduler should stop using a certain path when the RTT difference against the faster path is larger than a defined threshold. Both [58] and [112] present the decrease of download completion times, ranging from 20% to 40% depending on the path combinations. MPTCP-MA [113] uses MAC-layer information to estimate the path status of WiFi specifically and to possibly suspend its use during intermittent connectivity caused by the short signal range and susceptibility to interference. Experimental results show that MPTCP-MA can efficiently utilize an intermittently available path, with WiFi throughput improvements of up to 72%.

Finally, besides the scheduling algorithms reviewed above and categorized across different ATSSS steering modes, there are other works providing a multipath programming model or framework [93], [114]–[118]. In these references, the existing multipath scheduling algorithms exist as plugins and are called by the application via the provided Application Programming Interface (API).

C. MULTIPATH CONGESTION CONTROL

In line with its main design goals discussed in Section III-C, the multipath congestion control is in charge of a) avoiding harmful interaction with concurrent single-path traffic, and b) shifting traffic away from congested paths, thus load balancing and improving throughput. Therefore, among the different steering modes, congestion control can be considered partially as the Load-balancing steering mode in ATSSS.

The development of multipath congestion control algorithms, formally described in [69], dates back to 2005, when [119] explores the coupled CWND adjustment, in which all subflows belonging to the same multipath connection are adjusted simultaneously whether to increase or reduce their CWND. Then, Equally-Weighted TCP (EWTCP) [120] is proposed, which applies TCP NewReno on each MPTCP subflow independently, i.e., fairness to regular TCP is not the goal and each subflow is independent. To improve the performance of non-congested subflows, [69] proposes Linked-Increases Algorithm (LIA) which is a coupled congestion control where only the CWND of subflows experiencing congestion are reduced. However, it has been reported in [121], [122] that LIA could behave unfriendly towards regular TCP in some scenarios. Hence, Opportunistic Linked-Increases Algorithm (OLIA), which is also a coupled congestion control, is proposed in [121], [122]. OLIA explores the concept of Pareto optimality, i.e., the equilibrium of a resource allocation problem, where one flow cannot gain more resources without damaging the resources of other flows. From the experiments, OLIA is observed to let the single-path user to improve 2 times higher throughput than LIA. Meanwhile, OLIA significantly reduces the congestion level at the bottleneck, up to 6 times lower compared to LIA.

Along the same coupled congestion control design, [123] proposes Path Associativity Congestion Control (PACC), requiring that MPTCP subflows do not take a greater aggregate bandwidth than a single-path TCP flow on a shared bottleneck. Then, [124] studies a rate control to improve multipath transmission by simultaneously keeping fairness to regular TCP. The subflow rate of the proposed congestion control can be obtained from an optimization problem having TCP-friendliness as a required constraint. A parameterized formula to generalize the CWND update is given in [125]. In this approach, the optimization goal lies within a design space comprising fairness and responsiveness, where fairness can be sacrificed for higher responsiveness, resulting in higher throughput. Later, on the basis of [125], the work in [126] shows that OLIA can be unresponsive to network changes and proposes Balanced Linked Adaptation (BALIA). By generalising existing multipath congestion control algorithms, BALIA is able to dynamically balance the trade-off between friendliness to regular TCP and responsiveness thus complying to the coupled congestion control approach. The results show that, under the condition of guaranteeing the fairness, BALIA can still reach up to 4 times faster convergence time than OLIA.

Evaluated in server farms, [127] proposes One-ended multipath TCP (OmTCP), which modifies TCP's Selective Acknowledgment (SACK) option and fast retransmit mechanisms to adjust the sender rate to be fair to regular TCP. A TCP-friendly congestion control algorithm is proposed in [128] for the multipath Host Identity Protocol (mHIP). The work designs a two-level mechanism, which applies single-path additive-increase/multiplicative-decrease (AIMD) and a global congestion control that adjusts the aggressiveness of each connection against regular TCP in a shared bottleneck. Built upon the delay-based congestion control rather than the loss-based ones, [129] proposes weighted Vegas (wVegas), which uses packet queuing delay to infer congestion instead of packet loss and adjusts the subflow's CWND. Compared with LIA, wVegas is shown to be more sensitive to changes of network congestion and thus achieves more timely traffic shifting and quicker convergence. The experimental result also shows the improvement of fairness for wVegas, e.g., when wVegas is 12.3% off from the optimal fair allocation among two paths, LIA is 54.7% off from the optimal fair allocation.

Recently, [130] proposes a reinforcement learning scheme in multipath congestion control, i.e., SmartCC which takes an ACK as a reward and applies Q-learning to manage the CWND. SmartCC improves the median throughput of OLIA with 32%. However, this approach does not consider fairness to the regular TCP, i.e., it is an uncoupled congestion control approach. Considering fairness to the regular TCP, [131] employs the online convex optimization of the online learning to design the congestion control algorithm for MPTCP, named MPCC. Repeatedly, MPCC first selects the per-subflow rates and then receives the performance implication quantified by the utility function. The online convex optimization approach derives the per-subflow rates by aggregating the value of the utility function. Results show that MPCC provides an improvement (both in the mean and the median) of around 2.3 times in terms of file download speed over MPTCP with OLIA.

D. RELIABLE TRANSFER

To benefit either throughput or reliability and latency in multipath transport, adding encoded packets to the application data of the multipath transfer is proposed in combination with multipath schedulers. Such reliable transfer mechanisms exploit redundancy, which can be considered as the Redundant steering mode in ATSSS. However, while the redundant steering mode in ATSSS may already guarantee enough reliability for the application, e.g. by simply duplicating packets on more than one network access, the transport layer may or may not introduce reliability as part of its scheduling, i.e., potentially adding a second reliability level. As example of [132] mentions the drawback of MPTCP in ATSSS when carrying unreliable traffic, e.g., UDP, as it retransmits every lost packet leading to increased delay. Originally, the goal of redundancy in the transport layer was to avoid data retransmission over higher latency paths. In ATSSS, an additional redundancy level introduced by the transport layer could be explicitly used to map different packets, e.g., data or redundancy, onto different network accesses. This is however implementation specific. In the following, we enlist relevant literature in reliability applied to multipath transport without considering ATSSS.

Applying the coding upon base protocol of SCTP, [133] proposes eCMT-SCTP as the version of CMT-SCTP with erasure codes. Three different types of erasure codes are considered, i.e. block, convolutional and on-the-fly erasure codes integrated within CMT-SCTP. The evaluation targets generic web applications using fully reliable CMT-SCTP and video streaming using an equivalent of partially reliable CMT-SCTP. To further improve the performance, a modification of the SCTP retransmission mechanism is also proposed, with a variable retransmission delay (aRTX) based on the type of error correction code. Reference [133] shows that eCMT-SCTP can achieve from 10% to 80% improvements in application goodput than CMT-SCTP under lossy multipath network conditions with a minimal (10%) overhead due to the encoding-decoding process.

Applying the coding upon base protocol of TCP, [134] proposes Network Coding-based MPTCP (NC-MPTCP), which introduces NC to some of the subflows. The core idea is the mixed use of regular and network-coded subflows, where regular subflows deliver application data and network-coded subflows deliver linear combinations of application data. NC-MPTCP outperforms MPTCP with up to 26% upon lossy paths and performs similarly to MPTCP when the paths are homogeneous and at low loss rates. Multipath Loss-Tolerant (MPLOT) [135] exploits multipath diversity with erasure codes. MPLOT presents throughput aggregation in both homogeneous and heterogeneous multipath scenarios, outperforming the default MPTCP without erasure codes with up to 21.5%. Systematic Coding MPTCP (SC-MPTCP) [136] proposes to mitigate packet reordering for a constrained receive buffer, by proactively transmitting redundant packets. The redundant packets are continuously updated according to the estimated aggregate retransmission ratio.

Across experiments over paths with different heterogeneities and loss rates, SC-MPTCP can reach 3-8 times the average throughput compared to MPTCP. Coded TCP (C-TCP) [137] is implemented in user-space and only considers the case with two WLAN paths with systematic block codes. Further, C-TCP applies a modified version of congestion control, compared with what standard TCP applies, in two aspects: firstly, it takes both loss and delay as feedback signals instead of loss solely; secondly, it introduces a token to allow the sender to transmit a packet instead of CWND. Fountain code-based MPTCP (FMTCP) [138] exploits the random nature of the fountain code to flexibly transmit encoded symbols from the same or different data blocks over different subflows, which aims to mitigate the negative impact of the path heterogeneity. FMTCP demonstrates gains of more than 50% in aggregation over MPTCP are obtained in experiments with a non-shared bottleneck scenario. Reference [139] focuses on the experimental study of using NC over MPTCP in a car with cellular and WiFi links. A comparison between MPTCP and MPTCP/NC is presented using both the empirical data and mean-field approximation. The results show that network coding can provide users in mobile environments a higher quality of service, e.g., transmitting 100 times of packets per second than that of MPTCP when the lossy connection presents. QuAlity-Driven MultIpath TCP (ADMIT) [140] focuses on real-time high definition H.264 video using a MPTCP-model with FEC. It focuses on goodput, end-to-end delay and Peak Signal-to-Noise Ratio (PSNR) and presents the improved performance in these three aspects compared with not only MPTCP but also the aforementioned MPLOT and FMTCP. Stochastic Earliest Delivery Path First (S-EDPF) [141] integrates a novel low delay FEC scheme to increase the robustness of each channel and thereby minimizes the retransmission delay. Moreover, it models each path by considering the stochastic factor to increase the reliability of each decision. Reference [141] also reuses the framework of C-TCP [137] as introduced earlier. Reference [75] implements an XOR-based FEC within TCP, to aid multipath transport with heterogeneity. In such a case, the advantages of an XOR-based FEC approach are low computational overhead and simple implementation, where TCP's original segment structure can be maintained. However, the obvious disadvantage is that it can only recover one segment per block, e.g., if two or more packets are lost within a block the FEC packet is wasted.

Applying the coding upon base protocol of UDP or the combination of UDP and TCP, Bandwidth-Efficient Multipath Streaming (BEMA) [142] is built for H.264 video over multiple paths, considering quality metrics including video Peak Signal-to-Noise Ratio (PSNR), end-to-end delay, and goodput. Compared with FMTCP, BEMA improves PSNR by up to 23.3%, reduces end-to-end delay by up to 27.2%, and improves the goodput by up to 12.9%. BEMA uses UDP and TCP with TCP-Friendly Rate Control (TFRC) [143] and applies systematic Raptor codes and FEC adaptivity. Targeting towards QUIC, [144], [145] apply the use of FEC in MPQUIC. The experimental results indicate the performance increase compared with MPQUIC without FEC, especially in lossy networks. Similar to regular TCP, this can alleviate the burden of the congestion control layer of QUIC, especially in lossy networks where it is difficult to differentiate whether a loss is caused by link layer or congestion drop. Reference [146] targets multipath streaming protocols and builds upon its own multipath UDP. The work develops an analytical model and uses asymptotic analysis to derive a closed-form, optimal load splitting solution, based on the joint solution using FEC and multipath. Reference [147] proposes Multipath Multimedia Transport Protocol (MPMTP) and is constructed over both TCP and UDP flows. The TCP flow is used to exchange the control packets while the UDP flows are used to exchange the data. The work adapts the Raptor encoding parameters during the transfer, considering time-varying wireless networks and Raptor codes complexity. [148] targets the design of a strict time-critical transmission system using trace data from multipath UDP. FEC is used to optimize latency and reliability of the fixed-rate application traffic over channels with time-varying capacity. With the employed FEC mechanism, the reliability can be increased by up to 21.8% than the one without, depending on the code rate.

Finally, several proposals leverage the benefits of NC with MPTCP [149]–[154], however, they mainly focus on the integration of NC in the TCP level, not profiting from improvements from multipath transport.

E. MULTIPATH TRANSPORT AND 5G REQUIREMENTS

Based on the multipath transport literature surveyed through Sections IV-A, IV-B, IV-C, IV-D, and considering the mapping proposed with the ATSSS modes defined in Section II-E, we now summarize how the references fit into 5G requirements and, more specifically, how they could bring benefits to eMBB and/or URLLC services. The overview of the mapping between multipath literature, ATSSS modes, and 5G services are summarised in Table 2.

The use of multipath transport in 5G ATSSS is a clear enabler of eMBB and URLLC requirements addressing both high throughput and high reliability as well as low latency requirements. Generally speaking, multipath transport primarily supports bandwidth aggregation from different network paths, thus supporting eMBB slices to achieve higher throughput. In URLLC, slices benefit from multipath transport when scheduling policies based on shortest delay can take advantage of path redundancy, using the best lowest delay path currently available. The multipath transport solutions in Table 2 implemented as path management, scheduling, congestion control or reliable transfer algorithms can, in combination with the ATSSS modes in Section II-E bring benefits to eMBB and URLLC slices. In the following, while we focus on examples of multipath scheduling solutions applied to each of the ATSSS modes, we also highlight how other multipath solutions map to them.

TABLE 2. Mapping between multipath solutions, ATSSS modes, and 5G services (PM: Path Management, SCH: Scheduling, CC: Congestion Control, REL: Reliable Transfer).

Reference	Type	ATSSS Mode	Suggested 5G Service	Base Protocol	Evaluation	Year
[82]	PM	Active-Standby	eMBB/URLLC	SCTP	Simulations	2007
[87]	PM	Active-Standby	eMBB/URLLC	SCTP	Simulations	2009
[33]	PM	Active-Standby	eMBB/URLLC	TCP	Experiments	2012
[77]	PM	Active-Standby	eMBB/URLLC	TCP	Summary	2013
[78]	PM	Active-Standby	eMBB/URLLC	TCP	Simulations	2016
[79]	PM	Active-Standby	eMBB/URLLC	TCP	Simulations	2017
[99]	SCH	Load-balancing	eMBB	SCTP	Emulations	2011
[101]	SCH	Load-balancing	eMBB	SCTP	Simulations	2012
[102]	SCH	Load-balancing	eMBB	SCTP	Simulations	2013
[88]	SCH	Smallest Delay	eMBB / URLLC	TCP	IETF Standard	2013
[92]	SCH	Best-Access	eMBB / URLLC	TCP	Experiments	2013
[100]	SCH	Load-balancing	eMBB	SCTP	Simulations	2014
[58]	SCH	Active-Standby / Smallest Delay	eMBB / URLLC	TCP	Experiments	2014
[103]	SCH	Load-balancing	eMBB	TCP	Simulations	2014
[113]	SCH	Active-Standby / Smallest Delay	eMBB / URLLC	TCP	Experiments	2014
[89]	SCH	Best-Access	eMBB / URLLC	TCP	Emulations	2014
[155]	SCH	Priority	eMBB	SCTP	Simulations	2015
[112]	SCH	Active-Standby / Smallest Delay	eMBB / URLLC	TCP	Emulations	2015
[156]	SCH	Smallest Delay	eMBB / URLLC	TCP	Simulations	2015
[96]	SCH	Priority	eMBB / URLLC	TCP	Emulations / Experiments	2016
[93]	SCH	Best-Access	eMBB / URLLC	TCP	Simulations	2016
[107]	SCH	Redundant	URLLC	TCP	Emulations / Experiments	2016
[60]	SCH	Best-Access	eMBB / URLLC	TCP	Emulations / Experiments	2016
[108]	SCH	Redundant / Load-balancing	eMBB / URLLC	TCP	Emulations / Experiments	2017
[61]	SCH	Best-Access	eMBB / URLLC	TCP	Emulations / Experiments	2017
[70]	SCH	Smallest Delay	eMBB / URLLC	QUIC	Emulations	2017
[27]	SCH	Smallest Delay	eMBB / URLLC	QUIC	Emulations / Experiments	2018
[109]	SCH	Smallest Delay / Redundant	eMBB / URLLC	TCP	Emulations / Experiments	2018
[90]	SCH	Best-Access	eMBB / URLLC	QUIC	Emulations	2018
[91]	SCH	Best-Access	eMBB / URLLC	TCP	Simulations / Emulations	2018
[104]	SCH	Load-balancing	eMBB	TCP	Emulations / Experiments	2018
[110]	SCH	Redundant / Best Access	URLLC	TCP	Emulations / Experiments	2018
[111]	SCH	Best-Access / Redundant	eMBB / URLLC	TCP	Emulations	2019
[62]	SCH	Best-Access	eMBB / URLLC	TCP	Emulations / Experiments	2019
[63]	SCH	Best-Access	eMBB / URLLC	TCP	Emulations / Experiments	2019
[64]	SCH	Best-Access	eMBB / URLLC	QUIC	Emulations	2019
[97]	SCH	Priority	eMBB	TCP	Simulations	2019
[65]	SCH	Best-Access	eMBB / URLLC	QUIC	Emulation / Experiments	2020
[106]	SCH	Redundant / Best Access	eMBB / URLLC	QUIC	Emulations / Experiments	2020
[105]	SCH	Best-Access / Priority	eMBB / URLLC	QUIC	Emulations / Experiments	2020
[95]	SCH	Best-Access	eMBB / URLLC	QUIC	Emulations / Experiments	2021
[98]	SCH	Priority	eMBB	TCP	Emulations / Experiments	2021
[94]	SCH	Best-Access	eMBB / URLLC	TCP	Emulations / Experiments	2021
[119]	CC	Load-balancing	eMBB	TCP	Not Applicable	2005
[120]	CC	Load-balancing	eMBB	TCP	Simulations	2009
[127]	CC	Load-balancing	eMBB	TCP	Not Applicable	2010
[128]	CC	Load-balancing	eMBB	TCP	Simulations	2011
[123]	CC	Load-balancing	eMBB	TCP	Simulations	2011
[124]	CC	Load-balancing	eMBB	TCP	Simulations	2011
[26]	CC	Load-balancing	eMBB	TCP	Simulations / Experiments	2011
[129]	CC	Load-balancing	eMBB	TCP	Simulations	2012
[125]	CC	Load-balancing	eMBB	TCP	Simulations	2013
[121]	CC	Load-balancing	eMBB	TCP	Emulations / Simulations	2013
[126]	CC	Load-balancing	eMBB	TCP	Emulations	2016
[130]	CC	Not Applicable	eMBB	TCP	Simulations	2019
[131]	CC	Load-balancing	eMBB	TCP	Emulations / Experiments	2020
[135]	REL / SCH	Redundant / Best-Access	eMBB / URLLC	TCP	Simulations	2008
[146]	REL / SCH	Redundant / Load-balancing	eMBB / URLLC	TCP / UDP	Simulations	2009
[137]	REL / SCH	Redundant / Best-Access	eMBB / URLLC	TCP	Experiments	2012
[134]	REL / SCH	Redundant / Load-balancing	eMBB / URLLC	TCP	Simulations	2012
[139]	REL	Redundant	eMBB / URLLC	TCP	Simulations	2013
[133]	REL	Redundant	eMBB / URLLC	SCTP	Simulations	2013
[136]	REL	Redundant	eMBB / URLLC	TCP	Simulations	2013
[147]	REL / SCH	Redundant / Best-Access	eMBB / URLLC	TCP / UDP	Experiments	2014
[138]	REL / SCH	Redundant / Best-Access	eMBB / URLLC	TCP	Simulations	2015
[140]	REL / SCH	Redundant / Load-balancing	eMBB / URLLC	TCP	Emulations	2015
[142]	REL / SCH	Redundant / Best-Access	eMBB / URLLC	TCP / UDP	Emulations	2016

TABLE 2. *(Continued.)* **Mapping between multipath solutions, ATSSS modes, and 5G services (PM: Path Management, SCH: Scheduling, CC: Congestion Control, REL: Reliable Transfer).**

[141]	REL / SCH	Redundant / Best-Access	eMBB / URLLC	TCP	Emulations / Experiments	2017
[144]	REL / SCH	Redundant / Best-Access	eMBB / URLLC	QUIC	Emulations	2018
[148]	REL	Redundant	URLLC	UDP	Emulations / Experiments	2018
[75]	REL	Redundant	eMBB / URLLC	TCP	Emulations / Experiments	2019
[145]	REL / SCH	Redundant / Load-balancing	eMBB / URLLC	QUIC	Emulations	2019

Intuitively, Smallest Delay or Best-Access modes target 5G URLLC requirements. The Best-Access mode generalizes the Smallest Delay mode, since it targets the use of the access offering the best performance for a defined metric, which is still latency in many cases. However, by exploiting RTT together with other path characteristics, e.g., CWND, send window, RTT variation, etc., Best-Access can lead to enhanced scheduling policies, which can be beneficial for both eMBB and URLLC slices. On the other hand, assuming a multipath scheduler that takes a single metric into account, the goal of Best-Access might not be efficiently achieved. For example, disregarding path characteristics such as loss rates in combination with RTT, may not sufficiently address the requirements of URLLC or eMBB.

The Active-Standby mode foresees use cases combining this mode with others, e.g., Smallest Delay. As such, Active-Standby endows the Smallest Delay mode the ability to completely or periodically suspend the underperforming path, thus potentially leading to improved performance as compared to Smallest Delay alone. With the combination of Active-Standby and Smallest Delay modes, the goal of serving packets over the lowest latency path remains, which is beneficial for both eMBB and URLLC.

The Priority steering mode often foresees priority related to financial considerations at the user side, e.g., cost per bit sent on each path. Hence, a common approach is to set users' priority as the constraint for the optimization problem. The optimization goal can be however throughput-specific, which applies to eMBB, but also related to both latency and throughput, e.g., in deadline-aware video streaming applications, which thus maps to URLLC.

The Load-balancing mode balances traffic on the available accesses by sending a corresponding amount of packets or flows. An accurate load-balancing can maximize throughput and, in turn, improve eMBB performance. This applies not only to multipath scheduler solutions but also to the congestion control, which is mainly associated with eMBB in Table 2. The rationale is that one of the main objectives of congestion control is to maximize throughput while behaving friendly to other parallel connections (see Sections III-C and IV-C). We note however that current ATSSS specifications do not target a specific mode for congestion control.

Finally, the Redundant mode and reliable transfer mechanisms can be primarily adopted to meet high reliability requirements by URLLC services. How redundancy is utilized, plays a key role for meeting the expected latency and/or throughput performance. In this context, raw packet duplication over all paths may guarantee low latency while also enhancing reliability. However, this approach will result in significant overhead impacting the throughput. The overhead can be reduced by controlling the level of redundancy, i.e., duplication, which in turn will negatively affect reliability. In this view, FEC and NC approaches can deliver a better balance between throughput, latency and reliability by avoiding retransmissions and heavy redundancy overhead while still providing a certain degree of reliability. While most of the works in this category target throughput, some specifically control the redundancy degree to optimize for latency and reliability, see [135], [137], [138], [147]. Moreover, the combination of Redundant mode with other ATSSS modes can expand the applicability of the corresponding multipath solutions, i.e., Load-balancing combined with Redundant multipath schemes can be leveraged in both eMBB and URLLC.

V. OPEN RESEARCH ISSUES

The stringent requirements of 5G with high throughput, low latency and high reliability pose great challenges to research. Although incorporating multipath transport protocols in 5G is one of the solutions that targets and meets some of these requirements, we find some open research issues that deserve some attention. We summarize the key issues as follows.

A. EMERGING 5G APPLICATIONS

5G enables several new use cases compared to previous generations of cellular networks. While many works in the existing literature propose multipath transport solutions for improving *traditional* applications such as video streaming and web download, emerging 5G use cases, such as AR/VR, remote haptic control, autonomous driving and industrial remote control, have very different characteristics and requirements. For example, when interfacing with a robotic system, the requirements over each packet or flow could be different, e.g., packets could have distinct priorities due to different tasks and associated update frequency. Similar challenges are expected for other use cases, e.g., AR/VR, where traffic flows having different requirements, e.g., in terms of latency and reliability, are expected to be simultaneously exchanged in both downlink and uplink directions. Such complex systems are also often composed of a mixture of event- and time-driven tasks, where packet priorities and payload lengths can be less predictable. Therefore, how to exploit network access resources to address the requirements of these new applications while meeting critical system requirements, is a complex new challenge that needs further investigation.

B. TRANSPORT PROTOCOL FEATURES AND 5G MULTI-CONNECTIVITY

QUIC comes with *connection migration*, decoupling the transport layer connection from the underlying IP address, thus, seamlessly supporting the transition of a QUIC connection from one access network to another. Related to ATSSS, this feature natively supports Steering and Switching, i.e., a QUIC connection can be seamlessly moved from one network access to another. MPTCP can, on the other hand, support both ATSSS Splitting and Switching, delivering a smoother transition between network accesses, as a consequence of utilizing multiple paths simultaneously. Compared to QUIC, MPTCP can natively support more than a single ATSSS mode and serve use cases when more than a single network access is available and could be leveraged by the application. If MPQUIC is not adopted by ATSSS as envisioned, a QUIC-based solution is limited to Switching and Steering.

In the scenario where ATSSS supports Splitting, it is expected that the UE throughput increases. The underlying network access characteristics from frequency bands, e.g. mid- or high-bands, to transport layer characteristics, e.g. RTT, CWND, and packet loss rates, will determine how much higher the throughput can be. Due to the interaction of the radio with the environment or due to interference, it is not always the case that the promised high data rates can be guaranteed [95], and the support of additional paths can be crucial in such scenarios.

Apart from the recently adopted RFCs for QUIC at the IETF, there are several ongoing works around its future that are related to multi-connectivity. They include the multipath extension as well as an unreliable datagram extension [157], i.e., allowing traffic that does not need to be retransmitted. They also include evolving QUIC to be a generic tunnelling protocol for any type of traffic, i.e., not limited to a specific transport layer nor a specific protocol. We expect these developments to impact the future of ATSSS: from adoption of MPQUIC and thus supporting more than the basic *modes* and moving beyond the basic Steering, Splitting and Switching defined in Release 16 to enabling more flexible use cases and deployments.

C. POTENTIAL ADVANCES IN ATSSS

In the context of ATSSS Release 16, the modes discussed throughout this work largely focus on transport layer solutions using MPTCP. While ATSSS for Rel-17 (Phase 2) [16] and beyond (Phase 3) [43] are still ongoing work and having focus on QUIC and MPQUIC applied to ATSSS, we believe that there will be many opportunities to improve the performance and flexibility of ATSSS, once the proposed solutions are more settled and adopted. For example, it can be beneficial to consider the adoption of different multipath transport protocols based on their built-in features and upcoming extensions, specially in QUIC, for different scenarios. As specified in Release 16, ATSSS proposes the use of MPTCP to handle TCP traffic with a MPTCP proxy in the UPF, thus, excluding UDP and traffic from other layers such as IPsec and services such as Virtual Private Network (VPN). The ATSSS-LL (Lower Layer) function below IP is meant in this phase for this sort of traffic. In Release 17, ATSSS is meant to provide a more general transport solution to tunnel Ethernet frames over IP packets using QUIC, thus, including support for both TCP and UDP traffic. Here, if MPQUIC is adopted, it could support Splitting along with Steering and Switching. Tunneling non-TCP traffic over QUIC is possible, but as QUIC connections are fully encrypted and therefore cannot be intercepted, e.g., terminated at the UPF, a solution such as the ATSSS Release 16 with MPTCP is not possible. The added benefits and flexibility offered by QUIC, such as support for multi-streaming and more efficient loss recovery, motivate the interest in QUIC for a complementary generic and more flexible solution for ATSSS [14]. In this context, the above mentioned ongoing work at the IETF to design an unreliable datagram extension to QUIC for real-time data will be important, considering the large spectrum of new 5G applications and emerging URLLC use cases.

The current ATSSS modes are also very coarse-grained. For example, in the current conceptual description of steering modes, the smallest delay mode is logically included in the best-access mode. Additionally, FEC and NC are not considered part of the steering modes, even though it is shown they can provide great benefits towards low latency and reliability. Considering more fine-granular modes and extending the current modes to also cover congestion control and reliable transfer aspects will be of great importance moving forward. Finally, since ATSSS may have many different available plugins in terms of the transport protocol, scheduling, congestion control, FEC and NC, it will be important to consider how to integrate with or further develop the existing multipath programming models and frameworks introduced in Section IV-B.

D. MULTIPATH IN THE 5G NR ERA

Most of the existing multipath research is based on current cellular and local area networks and they tackle heterogeneous paths by looking at the mean value of path delay, loss rate, etc., while few others tackle the dynamicity of heterogeneous paths. However, 5G NR, specially at higher frequency bands, will inevitably bring higher dynamicity to the paths due to millimeter Wave (mmWave), line-of-sight requirements induced by beamforming or the handover between macro and small cells in more dense deployments. Therefore, future work needs to focus on such dynamicity by efficiently utilizing the statistical distribution of the path delay, loss, etc. To achieve this, we need to first better understand the path characteristic of 5G NR. To this end, experimental open-source 5G platforms such as mmFlex [158], and open source components such as openairinterface [159] that build on software-defined radio [160] will be crucial.

E. DATA-DRIVEN APPROACHES

Most of the existing multipath research apply mathematical modeling and optimization to design rule-based multipath algorithms. The advantage of these solutions is that they are often of low computation complexity, and the behavior of the algorithm they rely on is easily explainable. However, considering the dynamicity of the paths, especially in 5G as discussed above, such an approach might lack the ability to adapt to different path conditions.

Another approach is to unleash the power of data-driven algorithms in the multipath protocol design. The first trend towards this direction is to utilize the available labeled data from the transport layer itself to do classification and prediction. The second trend is, given the fact that real-world data are not labeled, to apply unsupervised learning (e.g., clustering) or reinforcement learning to derive appropriate policies. The advantage of data-driven solutions is that they have the potential to learn over different path conditions and accordingly adapt to them. However, they may lack explainability, which might be even more severe in URLLC, where reliability is difficult to mathematically prove or measure, i.e., you have 100% reliability until the first packet loss happens. Therefore, we argue that future research in this direction should bear the explainability point in mind. Moreover, data-driven solutions are normally of higher computation complexity compared to the rule-based counterpart, but this can be alleviated nowadays by using specialized hardware used for data-driven tasks.

F. MULTIPATH CONGESTION CONTROL FOR 5G

Several multipath congestion control algorithms were proposed with the notion of end-to-end network fairness, i.e., flows sharing a bottleneck must receive the same resource amount, e.g., bandwidth, from the network perspective. This has historically influenced the design and performance of multipath transport protocols such as MPTCP [161]. We argue that this particular fairness notion can be revisited for ATSSS for one main reason: The use of multiple access technologies, such as the combination of 3GPP and non-3GPP. ATSSS scenarios include in general different technologies, which often belong to distinct underlying network infrastructures, e.g., the non-3GPP access does not share the same radio base station as the 3GPP access.

Therefore, the strict assumption about shared bottlenecks may become less relevant compared to when the focus was on end-to-end Internet scenarios. The difference in communication patterns can be recalled from Figure 5 in Section II-D, where the core-centric integration path depicts the flows from both radio access technologies from the UE, i.e., cellular and WLAN, being aggregated into a single flow before leaving the cellular core network to the Internet. From the point where the flow leaves the cellular network, the notion of network fairness is still valid. In contrast, the above-the-core integration path depicts when both flows are transparent to the cellular core network, running end-to-end as a multipath flow.

VI. CONCLUSION

On the road to 5G, one of the design trends is moving towards aggregating multiple access networks. Multipath transport protocols, which exploit multiple network paths at the transport layer, play an essential role in such a design trend. We have presented what we believe to be the first survey of multipath transport protocols for 5G, subjecting to the standardized ATSSS architecture. With respect to this survey, we have reviewed the research efforts by academia and industry and the standardization efforts by 3GPP and IETF. Also, we have studied the characteristics of these works and, based on which, we have proposed their integration based on the ATSSS functionalities and steering modes, as well as suggesting their applicable 5G services. In addition to the foreseen benefits of incorporating multipath transport protocols into 5G, we also point out major existing research issues. We believe that our survey will serve as a guideline for future research works in applying multipath transport protocols for 5G and beyond.

REFERENCES

[1] *5G Network Architecture-A High Level View*. [Online]. Available: https://www.huawei.com/minisite/hwmbbf16/insights/5G-Nework-Architectur%e-Whitepaper-en.pdf

[2] M. Series, *IMT Vision–Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, document Recommendation ITU 2083, 2015.

[3] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, ''On load distribution over multipath networks,'' *IEEE Commun. Surveys Tuts.*, vol. 14, no. 3, pp. 662–680, 3rd Quart., 2011.

[4] A. L. Ramaboli, O. E. Falowo, and A. H. Chan, ''Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues,'' *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1674–1690, 2012.

[5] S. K. Singh, T. Das, and A. Jukan, ''A survey on internet multipath routing and provisioning,'' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2157–2175, 4th Quart., 2015.

[6] J. Qadir, A. Ali, K.-L. A. Yau, A. Sathiaseelan, and J. Crowcroft, ''Exploiting the power of multiplicity: A holistic survey of network-layer multipath,'' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2176–2213, 4th Quart., 2015.

[7] M. Li, A. Lukyanenko, Z. Ou, A. Ylä-Jääski, S. Tarkoma, M. Coudron, and S. Secci, ''Multipath transmission for the internet: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2887–2925, 4th Quart., 2016.

[8] C. Xu, J. Zhao, and G.-M. Muntean, ''Congestion control design for multipath transport protocols: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2948–2969, 4th Quart., 2016.

[9] M.-T. Suer, C. Thein, H. Tchouankem, and L. Wolf, ''Multi-connectivity as an enabler for reliable low latency communications—An overview,'' *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 156–169, 1st Quart., 2020.

[10] X. Lin, A. Grovlen, K. Werner, J. Li, R. Baldemair, J.-F.-T. Cheng, S. Parkvall, D. C. Larsson, H. Koorapaty, M. Frenne, and S. Falahati, ''5G new radio: Unveiling the essentials of the next generation wireless access technology,'' *IEEE Commun. Standards Mag.*, vol. 3, no. 3, pp. 30–37, Sep. 2019.

[11] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, ''What will 5G be?'' *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[12] M. Agiwal, A. Roy, and N. Saxena, ''Next generation 5G wireless networks: A comprehensive survey,'' *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[13] *: System Architecture for the 5G System*, Standard 23.501, v16.4, 3GPP, Mar. 2020.

[14] M. Boucadair, O. Bonaventure, M. Piraux, Q. De Coninck, S. Dawkins, M. Kuehlewind, M. Amend, A. Kassler, Q. An, N. Keukeleire, and S. Seo. (2020). *3GPP Access Traffic Steering Switching and Splitting (ATSSS) Overview for IETF Participants*. [Online]. Available: https://www.ietf.org/id/draft-bonaventure-quic-atsss-overview-00.txt

[15] *Study on Access Traffic Steering, Switch and Splitting Support in the 5G System (5GS) Architecture*, Standard 23.793, v1.1.0, 3GPP, Oct. 2018.
[16] *Study on Access Traffic Steering, Switch and Splitting Support in the 5G System (5GS) Architecture; Phase 2 (Release 17)*, Standard v17.0.0, 3GPP, Mar. 2021.
[17] R. Peter, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, and D. Aziz, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72–79, May 2017.
[18] *5G Enhanced Mobile Broadband*, Standard 26.891, v16, 3GPP, Dec. 2018.
[19] *Service Requirements for Next Generation New Services and Markets*, Standard 22.261, v15.5.0, 3GPP, Jul. 2018.
[20] *Study on Scenarios and Requirements for Next Generation Access Technologies*, Standard 38.913, v14.2.0, 3GPP, May 2017.
[21] *NR; Physical Layer Procedures for Control*, Standard 3GPP TS 38.213 v16.2.0, 2020.
[22] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 1970–1978.
[23] M. Alsenwi, N. H. Tran, M. Bennis, A. K. Bairagi, and C. S. Hong, "EMBB-URLLC resource slicing: A risk-sensitive approach," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 740–743, Apr. 2019.
[24] A. K. Bairagi, M. S. Munir, M. Alsenwi, N. H. Tran, S. S. Alshamrani, M. Masud, Z. Han, and C. S. Hong, "Coexistence mechanism between eMBB and uRLLC in 5G wireless networks," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1736–1749, Mar. 2021.
[25] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 951–964, Oct. 2006.
[26] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," in *Proc. NSDI*, vol. 11, 2011, p. 8.
[27] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath QUIC: A deployable multipath transport protocol," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.
[28] *Access to the 3GPP Evolved Packet Core (EPC) Via Non-3GPP Access Networks*, Standard 3GPP TS 24.302 v.10.4.0, 2011.
[29] D. Laselva, D. López-Pérez, M. Rinne, and T. Henttonen, "3GPP LTE-WLAN aggregation technologies: Functionalities and performance comparison," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 195–203, Mar. 2018.
[30] R. Bajracharya, R. Shrestha, R. Ali, A. Musaddiq, and S. W. Kim, "LWA in 5G: State-of-the-art architecture, opportunities, and research challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 134–141, Oct. 2018.
[31] Y.-C. Chen, Y.-S. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of MultiPath TCP performance over wireless networks," in *Proc. Conf. Internet Meas. Conf.*, Oct. 2013, pp. 455–468.
[32] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan, "WiFi, LTE, or both?: Measuring multi-homed wireless internet performance," in *Proc. Conf. Internet Meas. Conf.*, Nov. 2014, pp. 181–194.
[33] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/WiFi handover with multipath TCP," in *Proc. ACM SIGCOMM Workshop Cellular Netw., Oper., Challenges, Future Design (CellNet)*, 2012, pp. 31–36.
[34] Q. D. Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "A first analysis of multipath TCP on smartphones," in *Passive and Active Measurement*. Cham, Switzerland: Springer, 2016, pp. 57–69.
[35] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "Observing real smartphone applications over multipath TCP," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 88–93, Mar. 2016.
[36] Q. De Coninck and O. Bonaventure, "MultipathTester: Comparing MPTCP and MPQUIC in mobile environments," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2019, pp. 221–226.
[37] F. Fejes, S. Racz, and G. Szabo, "Application agnostic QoE triggered multipath switching for Android devices," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.
[38] S. K. Saha, A. Kannan, G. Lee, N. Ravichandran, P. K. Medhe, N. Merchant, and D. Koutsonikolas, "Multipath TCP in smartphones: Impact on performance, energy, and CPU utilization," in *Proc. 15th ACM Int. Symp. Mobility Manage. Wireless Access*, Nov. 2017, pp. 23–31.
[39] *Improving Network Reliability Using Multipath TCP*. Accessed: Jan. 5, 2020. [Online]. Available: http://developer.apple.com/documentation/foundation/urlsessionconfiguration/improving_network_reliability_using_multipath_tcp
[40] Q. An, Y. Liu, Y. Ma, and Z. Li. (Oct. 2020). *Multipath Extension for Quic*. Working Draft, IETF Secretariat, Internet-Draft Draft-an-Multipath-Quic-00. [Online]. Available: http://www.ietf.org/internet-drafts/draft-an-multipath-quic-00.txt
[41] J. Deutschmann, K.-S. Hielscher, and R. German. (Oct. 2020). *Multipath Communication With Satellite and Terrestrial Links*. Working Draft, IETF Secretariat, Internet-Draft draft-deutschmann-sat-ter-multipath-00. [Online]. Available: http://www.ietf.org/internet-drafts/draft-deutschmann-sat-ter-multipath%-00.txt
[42] N. Keukeleire, B. Hesmans, and O. Bonaventure, "Increasing broadband reach with hybrid access networks," *IEEE Commun. Standards Mag.*, vol. 4, no. 1, pp. 43–49, Mar. 2020.
[43] *Study on Access Traffic Steering, Switching and Splitting Support in the 5G System Architecture; Phase 3*, Standard 3GPP, Agenda Item 9.1.4, Oct. 2021.
[44] M. Amend, D. Hugo, A. Brunstrom, A. Kassler, V. Rakocevic, and S. Johnson. (Aug. 2021). *Dccp Extensions for Multipath Operation With Multiple Addresses*. Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tsvwg-multipath-dccp/00/
[45] E. Kohler, M. Handley, S. Floyd, and J. Padhye, *Datagram Congestion Control Protocol (DCCP)*, document RFC4340, Mar. 2006. [Online]. Available: https://www.hjp.at/doc/rfc/rfc4340.html
[46] J. Iyengar, K. Shah, P. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming," in *Proc. SPECTS*, 2004, pp. 951–964.
[47] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, *TCP Extensions for Multipath Operation With Multiple Addresses*, document RFC Editor, RFC 8684, Internet Requests for Comments, Mar. 2020, [Online]. Available: http://www.rfc-editor.org/rfc/rfc8684.txt
[48] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? Designing and implementing a deployable multipath TCP," in *Proc. USENIX NSDI*, 2012, pp. 399–412.
[49] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, and J. Bailey, "The QUIC transport protocol: Design and internet-scale deployment," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 183–196.
[50] J. Iyengar and M. Thomson, *Quic: A UDP-Based Multiplexed and Secure Transport*, document RFC Editor, RFC 9000, Internet Requests for Comments, May 2021.
[51] S. R. Das, "Evaluation of QUIC on web page performance," M.S. thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2014.
[52] K. L. Mcmillan and L. D. Zuck, "Formal specification and testing of QUIC," in *Proc. ACM Special Interest Group Data Commun.*, Aug. 2019, pp. 227–240.
[53] Q. Coninck and O. Bonaventure. (Nov. 2020). *Multipath Extensions for QUIC (MP-QUIC)*. Internet-Draft. [Online]. Available: http://www.ietf.org/internet-drafts/draft-deconinck-quic-multipath-06.t%xt
[54] Q. An, Y. Liu, Y. Ma, and Z. Li. (Oct. 2020). *Multipath Extension for QUIC*. Internet-Draft. [Online]. Available: https://tools.ietf.org/html/draft-an-multipath-quic-00
[55] Y. Liu, Y. Ma, C. Huitema, Q. An, and Z. Li. (Mar. 2021). *Multipath Extension for QUIC*. Working Draft, IETF Secretariat, Internet-Draft Draft-Liu-Multipath-Quic-03. [Online]. Available: https://www.ietf.org/archive/id/draft-liu-multipath-quic-03.txt
[56] Q. D. Coninck and O. Bonaventure. (Nov. 2020). *Multipath Extensions for QUIC (MP-QUIC)*. Internet Engineering Task Force, Internet-Draft Draft-Deconinck-Quic-Multipath-06. Work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-deconinck-quic-multipath-06
[57] O. Bonaventure, M. Piraux, Q. Coninck, M. Baerts, C. Paasch, and M. Amend. (Mar. 2020). *Multipath Schedulers*. Internet-Draft. [Online]. Available: http://www.ietf.org/internet-drafts/draft-bonaventure-iccrg-schedulers-%00.txt
[58] S. Ferlin, T. Dreibholz, and O. Alay, "Multi-path transport over heterogeneous wireless networks: Does it really pay off?" in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 4807–4813.
[59] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," in *Proc. ACM SIGCOMM Workshop Capacity Sharing Workshop*, Aug. 2014, pp. 27–32.

[60] S. Ferlin, O. Alay, O. Mehani, and R. Boreli, "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks," in *Proc. IFIP Netw. Conf. (IFIP Netw.) Workshops*, May 2016, pp. 431–439.
[61] Y.-S. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, "ECF: An MPTCP path scheduler to manage heterogeneous paths," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 147–159.
[62] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, O. Alay, and N. Kuhn, "Low-latency scheduling in MPTCP," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 302–315, Feb. 2019, doi: 10.1109/TNET.2018.2884791.
[63] H. Zhang, W. Li, S. Gao, X. Wang, and B. Ye, "ReLeS: A neural adaptive multipath scheduler based on deep reinforcement learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1648–1656.
[64] M. M. Rosello, "Multi-path scheduling with deep reinforcement learning," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2019, pp. 400–405.
[65] H. Wu, O. Alay, A. Brunstrom, S. Ferlin, and G. Caso, "Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2295–2310, Oct. 2020.
[66] M. Becke, T. Dreibholz, H. Adhari, and E. P. Rathgeb, "On the fairness of transport protocols in a multi-path environment," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 2666–2672.
[67] C. Raiciu, M. Handley, and D. Wischik, *Coupled Congestion Control for Multipath Transport Protocols*, document IETF RFC 6356, Tech. Rep., Oct. 2011.
[68] C. Raiciu, D. Wischik, and M. Handley, "Practical congestion control for multipath transport protocols," Univ. College London, London, U.K., Tech. Rep., 2009. Accessed: Jan. 10, 2020. [Online]. Available: http://nrg.cs.ucl.ac.uk/mptcp/mptcp-techreport.pdf
[69] C. Raiciu, M. Handley, and D. Wischik, *Coupled Congestion Control for Multipath Transport Protocols*, document RFC Editor, RFC 6356, Internet Requests for Comments, Oct. 2011, [Online]. Available: http://www.rfc-editor.org/rfc/rfc6356.txt
[70] Q. De Coninck and O. Bonaventure, "Multipath QUIC: Design and evaluation," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 160–166.
[71] J. Iyengar and I. Swett. (Jun. 2020). *QUIC Loss Detection and Congestion Control*. IETF Secretariat, Tech.Rep.draft-ietf-quic-recovery-29. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-quic-recovery-29.txt
[72] I. Johansson. (2016). *Congestion Control for 4G and 5G Access*. [Online]. Available: https://tools.ietf.org/html/draft-johansson-cc-for-4g-5g-02
[73] B. Adamson, C. Adjih, J. Bilbao, V. Firoiu, F. Fitzek, S. Ghanem, E. Lochin, A. Masucci, M.-J. Montpetit, M. Pedersen, G. Peralta, V. Roca, P. Saxena, and S. Sivakumar, *Taxonomy of Coding Techniques for Efficient Network Communications*, document RFC Editor, RFC 8406, Internet Requests for Comments, Jun. 2018.
[74] N. Kuhn, E. Lochin, F. Michel, and M. Welzl. (Mar. 2020). *Coding and Congestion Control in Transport*. Working Draft, IETF Secretariat, Internet-DraftDraft-irtf-nwcrg-coding-and-congestion-02. [Online]. Available: http://www.ietf.org/internet-drafts/draft-irtf-nwcrg-coding-and-congest%ion-02.txt
[75] S. Ferlin, S. Kucera, H. Claussen, and O. Alay, "MPTCP meets FEC: Supporting latency-sensitive applications over heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2005–2018, Oct. 2018.
[76] P. Saxena, T. Dreibholz, H. Skinnemoen, O. Alay, M. A. Vazquez-Castro, S. Ferlin, and G. Acar, "Resilient hybrid SatCom and terrestrial networking for unmanned aerial vehicles," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 418–423.
[77] M. A. P. Gonzalez, T. Higashino, and M. Okada, "Radio access considerations for data offloading with multipath TCP in cellular/WiFi networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2013, pp. 680–685.
[78] H. Sinky, B. Hamdaoui, and M. Guizani, "Proactive multipath TCP for seamless handoff in heterogeneous wireless access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4754–4764, Jul. 2016.
[79] S.-I. Sou and Y.-T. Peng, "Performance modeling for multipath mobile data offloading in cellular/Wi-Fi networks," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3863–3875, Sep. 2017.
[80] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath TCP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 266–277, 2011.
[81] L. Boccassi, M. M. Fayed, and M. K. Marina, "Binder: A system to aggregate multiple internet gateways in community networks," in *Proc. ACM MobiCom Workshop Lowest Cost Denominator Netw. Universal Access*, Sep. 2013, pp. 3–8.
[82] C.-M. Huang and C.-H. Tsai, "The handover control mechanism for multi-path transmission using stream control transmission protocol (SCTP)," *Comput. Commun.*, vol. 30, no. 17, pp. 3239–3256, Nov. 2007.
[83] Y. Cao, C. Xu, J. Guan, and H. Zhang, "Qos-driven SCTP-based multimedia delivery over heterogeneous wireless networks," *Sci. China Inf. Sci.*, vol. 57, no. 10, pp. 1–10, Oct. 2014.
[84] Y. Cao, Q. Liu, Y. Zuo, G. Luo, H. Wang, and M. Huang, "Receiver-assisted cellular/WiFi handover management for efficient multipath multimedia delivery in heterogeneous wireless networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 1–13, Dec. 2016.
[85] P. Soderman, K.-J. Grinnemo, G. Cheimonidis, Y. Ismailov, and A. Brunstrom, "An SCTP-based mobility management framework for smartphones and tablets," in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 1107–1112.
[86] Ł. Budzisz, J. Garcia, A. Brunstrom, and R. Ferrús, "A taxonomy and survey of SCTP research," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 1–36, Aug. 2012.
[87] L. Budzisz, R. Ferrus, F. Casadevall, and P. Amer, "On concurrent multipath transfer in SCTP-based handover scenarios," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–6.
[88] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," Internet Eng. Task Force, Tech. Rep. RFC 6824, Jan. 2013.
[89] F. Yang, Q. Wang, and P. D. Amer, "Out-of-order transmission for in-order arrival scheduling for multipath TCP," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2014, pp. 749–752.
[90] A. Rabitsch, P. Hurtig, and A. Brunstrom, "A stream-aware multipath QUIC scheduler for heterogeneous paths," in *Proc. Workshop Evol., Perform., Interoperability QUIC*, Dec. 2018, pp. 29–35.
[91] T. Shreedhar, N. Mohan, S. K. Kaul, and J. Kangasharju, "QAware: A cross-layer approach to MPTCP scheduling," in *Proc. IFIP Netw.*, May 2018, pp. 1–9.
[92] M. Coudron, S. Secci, G. Pujolle, P. Raad, and P. Gallard, "Cross-layer cooperation to boost multipath TCP performance in cloud networks," in *Proc. IEEE 2nd Int. Conf. Cloud Netw. (CloudNet)*, Nov. 2013, pp. 58–66.
[93] X. Corbillon, R. Aparicio-Pardo, N. Kuhn, G. Texier, and G. Simon, "Cross-layer scheduler for video streaming over MPTCP," in *Proc. 7th Int. Conf. Multimedia Syst.*, May 2016, p. 7.
[94] S. D. Sathyanarayana, J. Lee, J. Lee, D. Grunwald, and S. Ha, "Exploiting client inference in multipath TCP over multiple cellular networks," *IEEE Commun. Mag.*, vol. 59, no. 4, pp. 58–64, Apr. 2021.
[95] H. Wu, G. Caso, S. Ferlin, O. Alay, and A. Brunstrom, "Multipath scheduling for 5G networks: Evaluation and outlook," *IEEE Commun. Mag.*, vol. 59, no. 4, pp. 44–50, Apr. 2021.
[96] B. Han, F. Qian, L. Ji, and V. Gopalakrishnan, "MP-DASH: Adaptive video streaming over preference-aware multipath," in *Proc. 12th Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2016, pp. 129–143.
[97] K. Gao, C. Xu, J. Qin, L. Zhong, and G.-M. Muntean, "A stochastic optimal scheduler for multipath TCP in software defined wireless network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[98] M. Amend, V. Rakocevic, and J. Habermann, "Cost optimized multipath scheduling in 5G for video-on-demand traffic," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.
[99] F. H. Mirani, M. Kherraz, and N. Boukhatem, "Forward prediction scheduling: Implementation and performance evaluation," in *Proc. 18th Int. Conf. Telecommun.*, May 2011, pp. 321–326.
[100] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli, "DAPS: Intelligent delay-aware packet scheduling for multipath transport," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1222–1227.
[101] J. Wang, J. Liao, and T. Li, "OSIA: Out-of-order scheduling for in-order arriving in concurrent multi-path transfer," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 633–643, Mar. 2012.
[102] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: Quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2193–2205, Nov. 2013.
[103] D. Ni, K. Xue, P. Hong, and S. Shen, "Fine-grained forward prediction based dynamic packet scheduling mechanism for multipath TCP in lossy networks," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2014, pp. 1–7.

[104] H. Shi, Y. Cui, X. Wang, Y. Hu, M. Dai, F. Wang, and K. Zheng, ''STMS: Improving MPTCP throughput under heterogeneous networks,'' in *Proc. USENIX Conf. Usenix Annu. Tech. Conf. (USENIX ATC)*. Berkeley, CA, USA: USENIX Association, 2018, pp. 719–730. [Online]. Available: http://dl.acm.org/citation.cfm?id=3277355.3277425

[105] X. Shi, L. Wang, F. Zhang, B. Zhou, and Z. Liu, ''PStream: Priority-based stream scheduling for heterogeneous paths in multipath-QUIC,'' in *Proc. 29th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2020, pp. 1–8.

[106] V. A. Vu and B. Walker, ''On the latency of multipath-QUIC in real-time applications,'' in *Proc. 16th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)(8)*, Oct. 2020, pp. 1–7.

[107] A. Frommgen, T. Erbshauser, A. Buchmann, T. Zimmermann, and K. Wehrle, ''ReMP TCP: Low latency multipath TCP,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.

[108] Y. E. Guo, A. Nikravesh, Z. M. Mao, F. Qian, and S. Sen, ''Accelerating multipath transport through balanced subflow completion,'' in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 141–153.

[109] H. Lee, J. Flinn, and B. Tonshal, ''RAVEN: Improving interactive latency for the connected car,'' in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2018, pp. 557–572.

[110] B. Felix, I. Steuck, A. Santos, S. Secci, and M. Nogueira, ''Redundant packet scheduling by uncorrelated paths in heterogeneous wireless networks,'' in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 498–503.

[111] E. Dong, M. Xu, X. Fu, and Y. Cao, ''A loss aware MPTCP scheduler for highly lossy networks,'' *Comput. Netw.*, vol. 157, pp. 146–158, Jul. 2019.

[112] J. Hwang and J. Yoo, ''Packet scheduling for multipath TCP,'' in *Proc. 7th Int. Conf. Ubiquitous Future Netw.*, Jul. 2015, pp. 177–179.

[113] Y.-S. Lim, Y.-C. Chen, E. M. Nahum, D. Towsley, and K.-W. Lee, ''Cross-layer path management in multi-path transport protocol for mobile devices,'' in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2014, pp. 1815–1823.

[114] A. Frömmgen, A. Rizk, T. Erbshäußer, M. Weller, B. Koldehofe, A. Buchmann, and R. Steinmetz, ''A programming model for application-defined multipath TCP scheduling,'' in *Proc. 18th ACM/IFIP/USENIX Middleware Conf.*, Dec. 2017, pp. 134–146.

[115] P. Hurtig, S. Alfredsson, A. Brunstrom, K. Evensen, K.-J. Grinnemo, A. F. Hansen, and T. Rozensztrauch, ''A NEAT approach to mobile communication,'' in *Proc. Workshop Mobility Evolving Internet Archit.*, Aug. 2017, pp. 7–12.

[116] 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,'' 2019, *arXiv:1907.04567*.

[117] A. Nikravesh, Y. Guo, F. Qian, Z. M. Mao, and S. Sen, ''An in-depth understanding of multipath TCP on mobile devices: Measurement and system design,'' in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2016, pp. 189–201.

[118] C. Diop, G. Dugue, C. Chassot, and E. Exposito, ''QoS-oriented MPTCP extensions for multimedia multi-homed systems,'' in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 1119–1124.

[119] F. Kelly and T. Voice, ''Stability of end-to-end algorithms for joint routing and rate control,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 2, pp. 5–12, 2005.

[120] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda, ''Multipath congestion control for shared bottleneck,'' in *Proc. PFLDNeT Workshop*, vol. 357, 2009, p. 378.

[121] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, ''MPTCP is not Pareto-optimal: Performance issues and a possible solution,'' *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1651–1665, Oct. 2013.

[122] R. Khalili, N. Gast, and M. Popovic, ''Opportunistic linked-increases congestion control algorithm for MPTCP,'' Internet Eng. Task Force, Tech. Rep. Internet Draft draft-khalili-mptcp-congestion-control-05, Jul. 2013.

[123] Y. Liu, B. Wang, K. Xu, and Z. Ma, ''PACC: A path associativity congestion control and throughput model for multi-path TCP,'' *Proc. Comput. Sci.*, vol. 4, pp. 1278–1287, Jan. 2011.

[124] C. Pluntke and M. Rio, ''TCP-friendly rate control for non-TCP multipath flows,'' in *Proc. ACM CoNEXT Student Workshop CoNEXT Student*, 2011, p. 14.

[125] Q. Peng, A. Walid, and S. H. Low, ''Multipath TCP algorithms: Theory and design,'' *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 1, pp. 305–316, Jun. 2013.

[126] Q. Peng, A. Walid, J. Hwang, and S. H. Low, ''Multipath TCP: Analysis, design, and implementation,'' *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 596–609, Feb. 2016.

[127] I. van Beijnum, A. Azcorra, and M. Bagnulo, ''OmTCP: Increasing performance in server farms,'' in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–6.

[128] T. Polishchuk and A. Gurtov, ''Improving TCP-friendliness and fairness for MHIP,'' *Inforcommunications J.*, vol. 3, no. 1, pp. 1–9, 2011.

[129] Y. Cao, M. Xu, and X. Fu, ''Delay-based congestion control for multipath TCP,'' in *Proc. Adv. Multimedia, Commun. Netw.*, Dec. 2013, pp. 1–10.

[130] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, and S. Lu, ''SmartCC: A reinforcement learning approach for multipath TCP congestion control in heterogeneous networks,'' *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2621–2633, Nov. 2019.

[131] T. Gilad, N. Rozen-Schiff, P. B. Godfrey, C. Raiciu, and M. Schapira, ''MPCC: Online learning multipath transport,'' in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2020, pp. 121–135.

[132] M. Amend, E. Bogenfeld, A. Brunstrom, A. Kassler, and V. Rakocevic. (Jul. 2019). *A Multipath Framework for UDP Traffic Over Heterogeneous Access Networks*. Internet Engineering Task Force, Internet-Draftdraft-amend-tsvwg-multipath-framework-mpdccp-01. Work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-amend-tsvwg-multipath-frame%work-mpdccp-01

[133] G. Sarwar, P.-U. Tournoux, R. Boreli, and E. Lochin, ''ECMT-SCTP: Improving performance of multipath SCTP with erasure coding over lossy links,'' in *Proc. 38th Annu. IEEE Conf. Local Comput. Netw.*, Oct. 2013, pp. 476–483.

[134] M. Li, A. Lukyanenko, and Y. Cui, ''Network coding based multipath TCP,'' in *Proc. IEEE INFOCOM Workshops*, Mar. 2012, pp. 25–30.

[135] V. Sharma, S. Kalyanaraman, K. Kar, K. K. Ramakrishnan, and V. Subramanian, ''MPLOT: A transport protocol exploiting multipath diversity using erasure codes,'' in *Proc. IEEE INFOCOM - 27th Conf. Comput. Commun.*, Apr. 2008, pp. 121–125.

[136] M. Li, A. Lukyanenko, S. Tarkoma, Y. Cui, and A. Ylä-Jääskia, ''Tolerating path heterogeneity in multipath TCP with bounded receive buffers,'' *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 1, pp. 375–376, Jun. 2013.

[137] M. Kim, A. ParandehGheibi, L. Urbina, and M. Meedard, ''CTCP: Coded TCP using multiple paths,'' 2012, *arXiv:1212.1929*.

[138] Y. Cui, L. Wang, X. Wang, H. Wang, and Y. Wang, ''FMTCP: A fountain code-based multipath transmission control protocol,'' *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 465–478, Apr. 2015.

[139] J. Cloud, F. du Pin Calmon, W. Zeng, G. Pau, L. M. Zeger, and M. Medard, ''Multi-path TCP with network coding for mobile devices in heterogeneous networks,'' in *Proc. IEEE 78th Veh. Technol. Conf. (VTC Fall)*, Sep. 2013, pp. 1–5.

[140] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, ''Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks,'' *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2345–2361, Sep. 2016.

[141] A. Garcia-Saavedra, M. Karzand, and D. J. Leith, ''Low delay random linear coding and scheduling over multiple interfaces,'' *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3100–3114, Nov. 2017.

[142] J. Wu, C. Yuen, B. Cheng, Y. Yang, M. Wang, and J. Chen, ''Bandwidth-efficient multipath transport protocol for quality-guaranteed real-time video over heterogeneous wireless networks,'' *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2477–2493, Jun. 2016.

[143] S. Floyd, M. Handley, J. Padhye, and J. Widmer, ''TCP friendly rate control (TFRC): Protocol specification,'' Internet Eng. Task Force, Tech. Rep. RFC 5348, Sep. 2008.

[144] F. Michel, Q. De Coninck, and O. Bonaventure, ''Adding forward erasure correction to QUIC,'' 2018, *arXiv:1809.04822*.

[145] Q. De Coninck, F. Michel, M. Piraux, F. Rochet, T. Given-Wilson, A. Legay, O. Pereira, and O. Bonaventure, ''Pluginizing QUIC,'' in *Proc. ACM Special Interest Group Data Commun.*, Aug. 2019, pp. 19–24.

[146] A. L. H. Chow, H. Yang, C. H. Xia, M. Kim, Z. Liu, and H. Lei, ''EMS: Encoded multipath streaming for real-time live streaming applications,'' in *Proc. 17th IEEE Int. Conf. Netw. Protocols*, Oct. 2009, pp. 233–243.

[147] O. C. Kwon, Y. Go, Y. Park, and H. Song, ''MPMTP: Multipath multimedia transport protocol using systematic raptor codes over wireless networks,'' *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1903–1916, Sep. 2015.

[148] F. Gabriel, J. Acevedo, and F. H. P. Fitzek, ''Network coding on wireless multipath for tactile internet with latency and resilience requirements,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[149] S. Gheorghiu, A. L. Toledo, and P. Rodriguez, ''Multipath TCP with network coding for wireless mesh networks,'' in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.

[150] X. Zhuoqun, C. Zhigang, Y. Hui, and Z. Ming, "An improved MPTCP in coded wireless mesh networks," in *Proc. 2nd IEEE Int. Conf. Broadband Netw. Multimedia Technol.*, Oct. 2009, pp. 795–799.
[151] Z.-Q. Xia, Z.-G. Chen, Z. Ming, and J.-Q. Liu, "A multipath TCP based on network coding in wireless mesh networks," in *Proc. 1st Int. Conf. Inf. Sci. Eng.*, 2009, pp. 3946–3950.
[152] A. Kulkarni, M. Heindlmaier, D. Traskov, M.-J. Montpetit, and M. Médard, "An implementation of network coding with association policies in heterogeneous networks," in *Proc. Int. Conf. Res. Netw.* Berlin, Germany: Springer, 2011, pp. 110–118.
[153] J. Cloud, F. du Pin Calmon, W. Zeng, G. Pau, L. M. Zeger, and M. Medard, "Multi-path TCP with network coding for mobile devices in heterogeneous networks," in *Proc. IEEE 78th Veh. Technol. Conf. (VTC Fall)*, Sep. 2013, pp. 1–5.
[154] G. Giambene, M. Muhammad, D. K. Luong, M. Bacco, A. Gotta, N. Celandroni, E. K. Jaff, M. Susanto, Y. F. Hu, P. Pillai, and M. Ali, "Network coding applications to high bit-rate satellite networks," in *Proc. Int. Conf. Wireless Satell. Syst.* Cham, Switzerland: Springer, 2015, pp. 286–300.
[155] Y. Cao, Q. Liu, G. Luo, and M. Huang, "Receiver-driven multipath data scheduling strategy for in-order arriving in SCTP-based heterogeneous wireless networks," in *Proc. IEEE 26th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Aug. 2015, pp. 1835–1839.
[156] D. Ni, K. Xue, P. Hong, H. Zhang, and H. Lu, "OCPS: Offset compensation based packet scheduling mechanism for multipath TCP," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 6187–6192.
[157] T. Pauly, E. Kinnear, and D. Schinazi. (2019). *An Unreliable Datagram Extension to QUIC*. [Online]. Available: https://tools.ietf.org/html/draft-pauly-quic-datagram-04
[158] J. O. Lacruz, D. Garcia, P. J. Mateo, J. Palacios, and J. Widmer, "mm-FLEX: An open platform for millimeter-wave mobile full-bandwidth experimentation," in *Proc. 18th Int. Conf. Mobile Syst., Appl., Services*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–13.
[159] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A flexible platform for 5G research," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 33–38, 2014.
[160] F. K. Jondral, "Software-defined radio—Basics and evolution to cognitive radio," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 3, Dec. 2005, Art. no. 652784.
[161] D. Wischik, M. Handley, and M. B. Braun, "The resource pooling principle," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 5, pp. 47–52, Sep. 2008.

HONGJIA WU received the B.Sc. degree in automatic control from Northeastern University and the M.Sc. degree in embedded systems from TU Delft. He is currently pursuing the Ph.D. degree with Simula and OsloMet. His research interests include multipath protocols and robotic systems.

SIMONE FERLIN received the Dipl.-Ing. degree in information technology with major in telecommunications from Friedrich-Alexander Erlangen-Nuernberg University, Germany, in 2010, and the Ph.D. degree in computer science from the University of Oslo, Norway, in 2017. She is currently a Software Research Engineer in radio networks at Ericsson AB. Her research interests include intersection of cellular networks and the Internet, with a focus on computer networking, QoS and cross-layer design, transport protocols, congestion control, network performance, security, and measurements. Her dissertation focused on improving robustness in multipath transport for heterogeneous networks with MPTCP. She actively serves on technical boards of major conferences and journals in these areas.

GIUSEPPE CASO (Member, IEEE) received the Ph.D. degree from the Sapienza University of Rome, in 2016. From 2018 to 2021, he was a Postdoctoral Fellow with the MOSAIC Department, SimulaMet, Oslo, Norway. He was a Postdoctoral Fellow with the Sapienza University of Rome until 2018. From 2012 to 2018, he has held visiting positions at the Leibniz University of Hannover, King's College London, the Technical University of Berlin, and Karlstad University. He is currently an Experienced Researcher at Ericsson Research (Radio Systems and Standards), Kista, Sweden. His research interests include cognitive and distributed communications, resource allocation in cellular systems, the IoT technology and evolution, and location-based services.

ÖZGÜ ALAY (Member, IEEE) received the B.S. and M.S. degrees in electrical and electronic engineering from Middle East Technical University, Turkey, and the Ph.D. degree in electrical and computer engineering from the Tandon School of Engineering, New York University. Currently, she is an Associate Professor with the University of Oslo, Norway, and the Head of Department at the Mobile Systems and Analytics (MOSAIC), Simula Metropolitan, Norway. She is the author of more than 70 peer-reviewed IEEE and ACM publications and she actively serves on technical boards of major conferences and journals. Her research interests include mobile broadband networks, including 5G, low latency networking, multipath protocols, and robust multimedia transmission over wireless networks.

ANNA BRUNSTROM (Member, IEEE) received the B.Sc. degree in computer science and mathematics from Pepperdine University, CA, USA, in 1991, and the M.Sc. and Ph.D. degrees in computer science from the College of William and Mary, VA, USA, in 1993 and 1996, respectively. She joined the Department of Computer Science, Karlstad University, Sweden, in 1996, where she is currently a Full Professor and a Research Manager with the Distributed Systems and Communications Research Group. She has authored/coauthored over 170 international peer-reviewed journals and conference papers. Her research interests include internet architectures and protocols, techniques for low latency internet communication, multi-path communication, and performance evaluation of mobile broadband systems, including 5G.

Article II:

Wu, Hongjia, Özgü Alay, Anna Brunstrom, Simone Ferlin, and Giuseppe Caso. "Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments." *IEEE Journal on Selected Areas in Communications 38, no. 10 (2020): 2295-2310*. DOI: **10.1109/JSAC.2020.3000365**

[Article not attached due to copyright]

Article III:

Wu, Hongjia, Giuseppe Caso, Simone Ferlin, Özgü Alay, and Anna Brunstrom. "Multipath Scheduling for 5G Networks: Evaluation and Outlook." *IEEE Communications Magazine 59, no. 4 (2021): 44-50.*

DOI: **10.1109/MCOM.001.2000881**

[Article not attached due to copyright]

Article IV:

Wu, Hongjia, Özgü Alay, Anna Brunstrom, Giuseppe Caso, and Simone Ferlin. "FALCON: Fast and Accurate Multipath Scheduling using Offline and Online Learning." Submitted to *IEEE/ACM Transactions on Networking (2022)*

DOI: **https://doi.org/10.48550/arXiv.2201.08969**

FALCON: Fast and Accurate Multipath Scheduling using Offline and Online Learning

Hongjia Wu, Özgü Alay, Anna Brunstrom, Giuseppe Caso, Simone Ferlin

***Abstract*—Multipath transport protocols enable the concurrent use of different network paths, benefiting a fast and reliable data transmission. The scheduler of a multipath transport protocol determines how to distribute data packets over different paths. Existing multipath schedulers either conform to predefined policies or to online trained policies. The adoption of millimeter wave (mmWave) paths in 5th Generation (5G) networks and Wireless Local Area Networks (WLANs) introduces time-varying network conditions, under which the existing schedulers struggle to achieve fast and accurate adaptation. In this paper, we propose FALCON, a learning-based multipath scheduler that can adapt fast and accurately to time-varying network conditions. FALCON builds on the idea of *meta-learning* where offline learning is used to create a set of meta-models that represent coarse-grained network conditions, and online learning is used to bootstrap a specific model for the current fine-grained network conditions towards deriving the scheduling policy to deal with such conditions. Using trace-driven emulation experiments, we demonstrate FALCON outperforms the best state-of-the-art scheduler by up to 19.3% and 23.6% in static and mobile networks, respectively. Furthermore, we show FALCON is quite flexible to work with different types of applications such as bulk transfer and web services. Moreover, we observe FALCON has a much faster adaptation time compared to all the other learning-based schedulers, reaching almost an 8-fold speedup compared to the best of them. Finally, we have validated the emulation results in real-world settings illustrating that FALCON adapts well to the dynamicity of real networks, consistently outperforming all other schedulers.**

I. Introduction

The 5th Generation of mobile communications (5G) raises the expectations towards three key performance aspects: very high data rates, ultra-reliable and low-latency communications, and massive connectivity. To accommodate these requirements, the concurrent use of multiple Radio Access Technologies (RATs), i.e., *multi-connectivity*, is one of the key solutions highlighted in 5G systems [1]. Among several 5G multi-connectivity schemes [2], multipath transport protocols, such as multipath Transmission Control Protocol (MPTCP) [3] and multipath QUIC (MPQUIC) [4], have recently gained significant attention. In particular, this is due to the Technical Specification (TS) 23.501 (Release 16) by 3rd Generation Partnership Project (3GPP) [5], where it is discussed how 5G systems can take advantage of multipath transport protocols to support the Access Traffic Steering, Switching and Splitting (ATSSS) architecture, ultimately enabling multi-connectivity between 3GPP access, such as Long Term Evolution (LTE) and 5G New Radio (NR), and non-3GPP Wireless Local Area Networks (WLAN), such as WiFi.

Among the functionalities of multipath transport protocols, the multipath scheduler plays a key role since it regulates the distribution of data packets over different available paths (i.e., the available RATs), ultimately impacting the achievable performance in terms of experienced throughput, latency, and connection reliability. The design of a high-performing multipath scheduler is a challenging problem, especially under high time-varying network conditions, e.g., in the case of millimeter wave (mmWave) paths with high propagation losses and sensitivity to blockage [6]. To operate well in such challenging conditions, a multipath scheduler should be able to meet two main targets: a) *fast adaptation*, i.e., adapt its scheduling policy quickly to the network conditions, and b) *accurate adaptation*, i.e., the scheduling policy should capture the network conditions accurately.

Existing multipath schedulers are either based on predefined rules (e.g., using the path with minimum Round Trip Time (RTT)) or on Machine Learning (ML) schemes (e.g., using a Reinforcement Learning (RL) algorithm to select the best path to use under some specific network conditions). Schedulers based on predefined rules define a priori, rules based on the network conditions that they will adapt to (cf. Section II-B). As these schedulers do not need to *learn* the scheduling policy to use, their adaptation time is negligible meeting the fast adaptation target. However, the predefined rules often result in a coarse-grained scheduling policy that might not adapt well to the current network conditions, particularly when such conditions vary rapidly. Thus, schedulers based on predefined rules have difficulty in meeting the accurate adaptation target. Schedulers based on ML use, in particular, *online learning* approaches, observe the current network conditions and adapt to them by deriving a corresponding scheduling policy (cf. Sections II-B and II-C). Compared to schedulers based on predefined rules, they require extra time for *learning* the policy, thus resulting in slower but possibly more accurate adaptation to network conditions. In fact, a trade-off exists for these schedulers in terms of fast vs. accurate adaptation. On the one hand, if the scheduler (i.e., the learning agent) employs a complex learning architecture, e.g., a deep neural network [7], [8], it may converge to an accurate policy but this may require more time, thus inhibiting fast adaptation. On the other hand, if the scheduler employs a simple learning scheme, e.g., a lightweight RL algorithm [9], [10], it may converge faster at the cost of accuracy.

To address the above challenges faced by online learning schedulers, we argue that scheduling operations may benefit from further training based on *offline learning*. Indeed, a scheduler may use previous experience on already faced network conditions for deriving proper model(s) for newly encountered conditions; then, such model(s) can be exploited by the online learning algorithm for obtaining a fast and accurate scheduling policy. This idea is further clarified and justified throughout Sections II and III. Within the above context, this paper proposes FALCON, a ML-based multipath scheduler that combines online and offline learning. FALCON builds on the idea of *meta-learning* [11], [12], where a meta-model is set up via offline learning and fine-tuned via online learning. The online learning experience also feeds back to the offline learning function to form a closed loop for continuously updating the meta-model. The contributions of our work can be summarized as follows:

- We present the necessity for a multipath scheduler to be able to adapt fast and accurately to varying network conditions and show that existing multipath schedulers have difficulty to meet this objective;
- We design FALCON, an ML-based multipath scheduler that combines the benefits of offline and online learning for deriving trained multipath scheduling policies with a reduced amount of input data. To the best of our knowledge, our work is the first systematic study on multipath scheduling that optimizes both adaptation speed and accuracy to time-varying network conditions;
- We implement the protocol aspects of FALCON in MPQUIC using `quic-go` and the learning aspect of FALCON using `keras-rl`. All software components of FALCON are provided as open-source to the community.[1]
- Using trace-driven emulations, we demonstrate fast and accurate adaptation and thus the superior performance of FALCON for applications of bulk transfer and web service with multi-streaming support compared to the state-of-the-art multipath schedulers.
- We validate the emulation results in real-world settings and show that FALCON outperforms all other schedulers in realistic network conditions.

The rest of this paper is organized as follows. We first summarize the foundations and related work of our work in Section II. We then specify the research problem and provide an overview of FALCON in Section III. We next detail the design of FALCON in Section IV. We present the experimental setup in Section V and evaluate the performance of FALCON via emulations in Section VI and real-world experiments in Section VII. We finally conclude our work in Section IX.

II. Foundations and Related Work

In this section, we summarize foundations and related work of FALCON, including aspects related to multipath transport (Section II-A), multipath scheduling (Section II-B), and learning in networking scenarios (Section II-C).

A. Multipath Transport Protocol

Multipath transport protocols are designed to achieve higher throughput and resilience compared to their single-path counterparts, since they can leverage several paths simultaneously and support seamless failover. In particular, two multipath protocols have wide support from both standardization and research communities: MPTCP and MPQUIC.

MPTCP [3] is the multipath extension of TCP and has the goal of being transparent to both higher and lower protocol layers. Its design and operation are influenced by the proliferation of middleboxes, meddling in end-to-end TCP connections, and preventing TCP extensions as well as the deployment of new transport protocols. Adopting several successful features of TCP, QUIC became recently an attractive alternative, as it integrates Transport Layer Security (TLS) and improves latency at the connection start. Differently from TCP, QUIC encrypts most of the protocol headers and all payloads to prevent interference from middleboxes. Motivated by the success of MPTCP, there are already some MPQUIC implementations proposed as multipath extensions of QUIC [4], [13]. We leverage MPQUIC to perform the analysis of multipath schedulers in this paper, as we believe it will play a key role in determining the multi-connectivity performance in 5G.

B. Multipath Scheduling

The multipath scheduler is in charge of distributing packets over the available paths. In the following, we describe two categories of multipath schedulers: Based on predefined rules and based on ML schemes.

Schedulers based on predefined rules: Traditional multipath schedulers follow predefined rules that do not change over time. For example, a Round Robin (RR) scheduler cyclically sends packets over each path, as long as there is space in the congestion window (CWND) of the paths. RR may perform reasonably well when the available paths have similar characteristics (i.e., paths are *homogeneous*). However, since it does not consider the characteristics of the individual paths it is unable to prevent out-of-order packet arrival at the receiver, which is detrimental to multipath transport performance. The minimum RTT (minRTT) scheduler has shown that considering and exploiting path characteristics, e.g., by sending packets on the path with available CWND and lowest RTT, allows achieving higher throughput [14]. Indeed, minRTT is the default scheduler in both MPTCP and MPQUIC.

Other schedulers based on predefined rules have been proposed over the years. Blocking Estimation (BLEST) [14] and Earliest Completion First (ECF) [15] try to provide both high throughput and low latency. Assuming two available paths, when both paths have CWND availability, BLEST and ECF behave like minRTT, i.e., they select the path with the lowest RTT. When the path with the lowest RTT has no CWND availability, BLEST and ECF use different mechanisms to decide whether it is better to send packets on the path with the highest RTT or wait for the path with the lowest RTT to become available again. Addressing specific use cases and applications, the works in [16], [17], [18] apply an adaptive packet duplication mechanism to guarantee robustness, which proves to be effective when extra data usage and battery consumption are not limiting factors. The work in [19] proposes the Slide Together Multipath Scheduler (STMS) to reduce out-of-order packet arrivals and, thus, the receiver buffer problem.

[1] Upon acceptance of the paper.

[20] proposes a loss-aware scheduler targeting networks with more than 20% loss rates. [21] proposes the Short Transfer Time First (STTF) scheduler, targeting low latency for short transfers and considering TCP specific aspects such as the TCP Small Queues (TSQ). Lastly, [22] proposes a multipath scheduler for MPTCP that targets IEEE 802.11 ad/ac WLANs.

Schedulers based on ML: Nevertheless, confronted with the complexity of the network conditions, it is difficult for schedulers based on predefined rules to guarantee the accuracy for various environment characteristics. Multipath scheduling can be also thought of as a decision-making problem thus naturally fitting into scenarios that RL schemes aim to solve, including multi-armed bandit problems (MAB) and Markov decision processes (MDP). For this reason, there is an emerging interest in developing ML-based multipath schedulers. Adopting the MAB framework, [9] combines the Linear Upper Confidence Bound (LinUCB) algorithm and a stochastic adjustment to design a multipath scheduler in MPQUIC, namely Peekaboo, that shows improved performance in dynamic heterogeneous networks compared to schedulers based on predefined rules. Then, [10] proposes Modified-Peekaboo (M-Peekaboo) by extending the learning scheme of Peekaboo for path selection, aiming at extending the applicability range towards 5G mmWave networks. Framing the scheduling problem as an MDP, [7] exploits the Deep Q-Network (DQN) architecture to design a multipath scheduler in MPTCP, namely Reles, which shows performance gains over minRTT. Similarly, [8] also designs a multipath scheduler using DQN in MPQUIC, resulting in no clear performance gain over minRTT.

C. Learning Concepts in Networking

As mentioned in Section I, our proposed scheduler, FALCON, belongs to the category of schedulers based on ML. However, as also clarified later, we aim at not only leveraging previous scheduling approaches, all based on online learning but also to include offline learning, to improve the overall performance. Hence, in this section, we provide an overview of both offline and online learning approaches currently considered and adopted in networking applications more generally, thus, not limited to multipath scheduling. Then, we also provide a high-level description of meta-learning, which is the actual framework used in FALCON for leveraging offline and online learning functionalities.

Offline learning: This paradigm assumes that, in order to derive a model of and/or a policy for a generic environment, an ML algorithm uses environment characteristics, i.e., data, collected well-ahead, before the derived model is meant to be used. In the following, we refer to pre-collected data as *offline data*. The learning outcome, e.g., the policy to be used by a network protocol, is not modified once derived on offline data. In other words, there is no retraining. Therefore, the assumption is that offline data includes a complete enough set of environment characteristics that could be experienced when the model/policy is actually used. To mention a few, offline learning is used to derive offline data-based policies for congestion control using optimization approach [23], Adaptive Bit Rate (ABR) streaming using DQN [24] or Asynchronous Advantage Actor Critic (A3C) [25], and device resource management using DQN [26] or Support Vector Machine (SVM) [27]. To the best of our knowledge, offline learning is not currently used for multipath scheduling.

Online learning: This paradigm assumes that to derive a model and/or policy, an ML algorithm uses data that is collected while the model/policy is being derived and used. In the following, we refer to run-time collected data as *online data*. Differently from the offline learning paradigm, the learning outcome is thus modified and adapted at run-time, exploiting newly encountered environment characteristics, i.e., new online data. This is commonly performed via two main approaches, i.e., with or without the use of an *abandoning mechanism*. In the first approach, the model/policy is abandoned when either a significant change in the environment characteristics is detected via so-called *change point detection* [9], [28], or a predefined timer expires [29], [30], [31]. Peekaboo [9] and M-Peekaboo [6] are relevant examples of schedulers that use an online learning approach with the change point detection. In the second approach, the online learning algorithm does not apply the abandoning mechanism, i.e., the model/policy is continuously updated since the algorithm is continuously fed with online data [32]. Hence, in this case, there is no abrupt model/policy abandoning, which may cause a slower reaction to sudden changes in the environment characteristics. Examples of multipath schedulers that use online learning with no abandoning mechanisms are [8], [7]. It is worth mentioning that the aforementioned online learning approaches may face the well-known catastrophic forgetting problem [33]. Indeed, with or without abandoning mechanisms, the continuous feed of online data may result in the derivation of new models/policies; old models/policies that resulted to be optimal for specific environment characteristics may thus be discarded, and they need to be re-discovered if the same environment characteristics reappear. As a remedy, for example, [34] tries to apply lifelong learning for video streaming to alleviate the catastrophic forgetting problem.

Meta-learning: The meta-learning paradigm, also known as "learning to learn" [35], combines online and offline learning. The goal of meta-learning is to derive (offline) a so-called meta-model for the set of learning tasks an ML algorithm needs to solve. The meta-model is built so that it can be rapidly adapted (online) to any new learning task that may be encountered, exploiting just a few experiences from the new task. The works in [11], [12] validate a meta-learning framework that can be used in several learning tasks, e.g., it can be applied to both supervised ML (regression and classification) and RL scenarios. Other works propose meta-learning for more specific scenarios, i.e., the update rule and selective copy of weights of deep networks [36], [37], [38] and recurrent networks [39], [40], [41]. In this paper, we design FALCON based on meta-learning paradigm to obtain fast and accurate scheduling policies.

III. Problem statement and solution overview

In this section, we explain the research problem (Section III-A) and provide an overview of our solution (Section III-B).

A. Problem Statement

The network conditions faced by a multipath scheduler vary in time, due to network congestion, users' mobility, dynamic characteristics of the wireless channels, etc. The recent use of mmWave spectrum in cellular networks and WLANs further increases this variability [42], [43]. Hence, as highlighted in Section I, a multipath scheduler should be able to adapt fast and accurately to challenging time-varying network conditions. Upon detection of a change of the network condition by the scheduler, adapting fast indicates that the adaptation time for realizing the adapted policy should be as small as possible; adapting accurately indicates that the adapted policy should match the current network condition as much as possible. This is, however, a difficult task, and *the design of a multipath scheduler that can adapt fast and accurately to time-varying network conditions* is an open research problem.

In the following, we clarify the limitations of existing schedulers (based on either predefined rules or learning paradigms) in meeting the above objective. Then, we also analyze the limitations that schedulers based on a pure offline learning approach would face. This analysis serves for further motivating our approach in designing FALCON, summarized in Section III-B and detailed in Section IV, where we combine the benefits of offline and online learning approaches.

Schedulers based on predefined rules can adapt fast but not accurately to time-varying network conditions. This is due to the inherent limitation caused by predefining the rule to follow for scheduling packets over the available paths. Indeed, the rule is usually rather simple and coarse-grained (e.g., select the path with minimum average RTT), thus failing to adapt accurately to the complex dynamics of the network conditions.

Schedulers based on online learning can ensure the derivation of an accurate scheduling policy. In general, however, the need for learning the network conditions online makes the adaptation slower compared to schedulers based on predefined rules. In order to speed up adaptation, schedulers based on online learning can sacrifice accuracy, thus exploiting a limited amount of data (observed network conditions) and a simple learning architecture for deriving a policy. In the following, we refer to these schedulers as *Type-I* online learning based schedulers. As empirically shown in [10], state-of-the-art Type-I schedulers still face challenges in satisfying the requirements in terms of adaptation time of modern networks, e.g., 5G mmWave. If accurate adaptation is preferred over fast adaptation, the online scheduler can exploit a larger amount of data and a more complex learning model. In the following, we refer to these schedulers as *Type-II* online learning based schedulers.

Schedulers based on offline learning may intuitively seem like a reasonable approach for achieving both fast and accurate adaptation. An offline learning-based scheduler may adapt fast because it is pre-trained. Moreover, such a scheduler might achieve accurate adaptation if trained on all the possibly encountered network conditions. However, this assumption is rather unrealistic for two main reasons: (1) Collecting all possible network conditions (past and future) is nearly impossible [44], [45]; (2) Even if all combinations of network conditions could be found, it is difficult to accurately label each of them mathematically. Hence, several combinations of network conditions may be involved in the pre-training, and the obtained model would still have a coarse-grained match with the fine-grained network conditions.

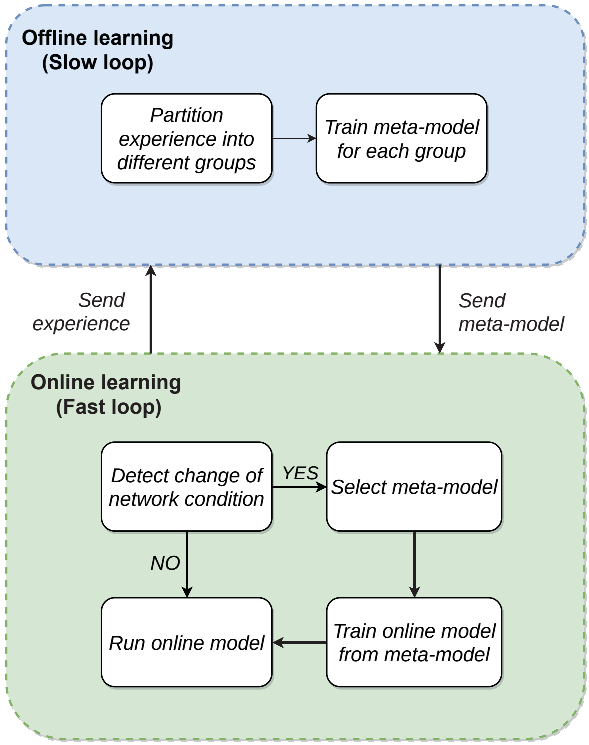


Figure 1. An illustration of the FALCON architecture.

B. Solution Overview

In this work, we design and implement a scheduler based on learning. We make this choice since, compared to schedulers based on predefined rules, schedulers based on learning have the ability to learn from the encountered network conditions and adapt to their variability over time.

To solve the research problem and overcome the limitations of online-only and offline-only learning-based schedulers outlined above, we propose FALCON, a multipath scheduler that combines offline learning and online learning. The key idea in FALCON is to use the meta-learning framework as offline learning to create a set of meta-models that represent the network conditions. Then, the set of meta-models is used by an online learning algorithm to bootstrap a specific model for the current network conditions and derive the scheduling policy to deal with such conditions. The meta-models are created so that they can converge to any specific model with only a small amount of online data.

On the one hand, the set of meta-models is a common root for the specific models. It is a sort of global view and thus, in contrast to the specific models, it is not very sensitive to the variability of network conditions. Hence, it can be updated at a relatively slow rate. On the other hand, online learning performs a fine-tuning of the meta-models and eventually obtains the specific model that suits the current network conditions. Hence, online learning operations are performed at a faster rate so to cater to the changes in network conditions. In other words, the model update is split into low-frequency and high-frequency updates.

Compared to the offline-only learning approach, FALCON has the ability to adapt to the current environment without labeling the current network conditions specifically to avoid the issues of both dealing with unseen network conditions and matching a coarse-grained model to a fine-grained network condition. Compared to Type-I online learning based schedulers, FALCON efficiently uses more data and a refined learning architecture, thus achieving a higher accuracy without sacrificing fast adaptation. Compared to Type-II online learning based schedulers, the loop of creation and refinement of meta-models allows achieving faster adaptation without sacrificing accuracy. Figure 1 illustrates the FALCON architecture whose main functions can be summarized as follows:

Offline Learning: Based on the experiences from the online learning module, the offline learning module partitions the experiences into different groups, based on the network conditions. For each group of experiences, the offline learning module performs meta-learning and derives a meta-model. The offline learning is set up tackling the changes of meta-model in real-world scenarios. The shared knowledge stored in the meta-model does not change in a very fast manner but rather an extremely gradual manner to slightly adjust with the internal representation in the real-world, thus updating in a very slow frequency.

Online Learning: The online learning module continuously monitors the change of network conditions. Depending on the outcome of the change detection, the online learning module may choose to deploy the current model (no changes are detected) or perform the model retraining (a change is detected). In the second case, the online learning module performs a training from a meta-model selected based on the group where the current network condition belongs. Thus, the online learning loop updates in a fast frequency.

Information Exchange: Online and offline modules cooperate and exchange experiences and meta-models in a recursive loop, as also shown in Figure 1.

IV. FALCON Design

In this section, we describe FALCON's algorithm by presenting its pseudo-code and the learning strategies adopted in the offline and online learning modules (Section IV-A). Then, we further specify the learning elements needed for setting up FALCON operations (Section IV-B).

A. Algorithm

Algorithm 1 reports FALCON pseudo-code. As also illustrated in Figure 1, FALCON leverages both offline and online learning via dedicated modules that exchange current experience and meta-models in a recursive loop. In the following, we provide more details on both modules.

Offline learning module: This module derives a set of meta-models that represent, on a high-level, the network conditions under which FALCON operates. The meta-models enable the online learning module to timely derive an accurate scheduling policy for the current network conditions. To do so, FALCON leverages the concept of meta-learning, where the main idea is to find a meta-model, denoted $\mathbf{\Theta}$, for solving a generic learning *task*. Meta-model $\mathbf{\Theta}$ represents the common starting point from which a number of refined models, that map onto more specific learning tasks, can be derived. For example, a meta-model is a high-level knowledge that a RL agent may have on how to navigate mazes (i.e., a generic task). Then, when the agent is deployed in a maze with specific characteristics (i.e., a specific task), it can exploit the high-level knowledge so to quickly learn how to navigate that specific maze [11]. Indeed, $\mathbf{\Theta}$ is created so that the refined models that match specific tasks can be derived in a few gradient steps. The requirement of $\mathbf{\Theta}$ is that starting from $\mathbf{\Theta}$, the online model can converge within several online gradient steps to match the presented network condition. In other words, $\mathbf{\Theta}$ guarantees that a few-shot learning [46], [47] is sufficient for finding the refined models. Considering that one online model may have several convergence points within the parameter space subject to the common machine learning paradigm, $\mathbf{\Theta}$ ensures the convergence points of different online models are close by each other.

Assuming to have a distribution of specific tasks, the derivation of meta-model $\mathbf{\Theta}$ follows this general procedure:

1) Initialize $\mathbf{\Theta}$;
2) Randomly sample a task from the task distribution;
3) Perform K steps of gradient descent updates on the task, starting from $\mathbf{\Theta}$, so to obtain a new representation, denoted $\boldsymbol{W}$;
4) Update $\mathbf{\Theta}$, that is, $\mathbf{\Theta} \leftarrow \mathbf{\Theta} + \lambda(\boldsymbol{W} - \mathbf{\Theta})$, where λ is the learning rate ($0 < \lambda \leq 1$);
5) Repeat steps $2-4$ until $\mathbf{\Theta}$ is found to be optimal by the adopted optimization routine.

Algorithm 1 FALCON pseudo-code.

Input:
1) T_{upd}: Update interval of meta-models;
2) $\boldsymbol{R}_{\mathsf{S}}$: Set of network condition ranges for meta-models;

Offline learning module:

while True **do**
  $\boldsymbol{Exp}$ = CollectOnlineExperience();
  $\mathbf{\Theta}_{\mathsf{S}}$ = MetaLearn($\boldsymbol{Exp}, \boldsymbol{R}_{\mathsf{S}}$);
  Wait(T_{upd});
end while

Online learning module:

while True **do**
  $\mathbf{\Theta}_{\mathsf{S}}$ = CollectOfflineMetaModels();
  $\boldsymbol{Exp}$ = Execute(**CurrentPolicy**);
  DetectedChange = ChangeDetect();
  if DetectedChange **then**
    $\mathbf{\Theta}$ = SelectMetaModel($\boldsymbol{R}_{\mathsf{S}}, \mathbf{\Theta}_{\mathsf{S}}$);
    NewPolicy = FewShotLearn($\mathbf{\Theta}$);
    CurrentPolicy ← **NewPolicy**;
  end if
end while

Once derived with the above procedure, meta-model $\boldsymbol{\Theta}$ can be used as a starting point for finding any specific model that suits a newly encountered task, by only using a small amount of experience collected on this new task [11], [12].

In our scenario, the learning tasks of FALCON are the different network conditions it may encounter and it should adapt to by deriving specific scheduling policies. In particular, we consider packet loss rate, mean RTT, and RTT variation rate of the available paths as indicators of network conditions. Moreover, FALCON adopts a DQN in the online learning module to derive its scheduling policies. Therefore, $\boldsymbol{\Theta}$ is defined as the initial set of parameters of the deep neural network used by DQN in the online learning module. Among common gradient descent approaches, we apply the mini-batch gradient descent rather than the stochastic gradient descent to cater for the use of DQN.

Note that the creation of a unique meta-model for all possible network conditions may require a significant increase of the number of gradient steps (K) needed to converge to an optimal $\boldsymbol{\Theta}$. Therefore, FALCON does not create a unique meta-model for representing all network conditions, but instead creates a set of meta-models, i.e., $\boldsymbol{\Theta}_{\mathsf{S}}$ (subscript S stands for set). Each meta-model in $\boldsymbol{\Theta}_{\mathsf{S}}$ is then created so to cover only a partial range of possible network conditions. For example, assuming to have two available paths, the x-th meta-model in $\boldsymbol{\Theta}_{\mathsf{S}}$, i.e., $\boldsymbol{\Theta}_x$, covers the range where, on path 1, packet loss rate is between $[a, b]$ %, mean RTT is between $[c, d]$ ms, and RTT variation rate is between $[e, f]$ %, and similar bounds are defined for path 2.

The number of meta-models and the ranges of network conditions covered by different meta-models are predefined, as reported in Section V. In the following, $\boldsymbol{R}_{\mathsf{S}}$ denotes the set of ranges on which the meta-models operate. As shown in Algorithm 1, the offline learning module updates $\boldsymbol{\Theta}_{\mathsf{S}}$ with a predefined update interval, i.e., T_{upd}. To do so, it first collects experience on the currently deployed policy (state, action, reward, as defined in Section IV-B) and on current network conditions (packet loss rate, mean RTT, and RTT variation rate), denoted $\boldsymbol{Exp}$, from the online learning module. Then, using the set of network conditions in $\boldsymbol{Exp}$, and comparing them with $\boldsymbol{R}_{\mathsf{S}}$, the offline learning module updates the corresponding meta-models in $\boldsymbol{\Theta}_{\mathsf{S}}$, following the procedure described above.

Online learning module: This module runs continuously for deriving the scheduling policies to use under different network conditions. FALCON uses a change point detection mechanism to trigger the selection of the meta-model that covers the new conditions, and the derivation of a new policy leveraging the selected meta-model.

Change point detection is thus an important aspect in FALCON, and it is particularly important in the wireless scenarios it faces, since these scenarios often result in high dynamicity and network changes, e.g., handovers. Intuitively, one could fix a detection interval and monitor the statistics of network conditions in such an interval. Then, if the difference of statistics exceeds a threshold, a change in network condition is detected. How to setup the detection interval is, however, not trivial: if the interval is too short, the change detection may be affected by short-term noise; if it is too long, actual changes might be lost. A similar problem exists for setting up the threshold that identifies an actual change in network conditions. In short, a hard-coded setup of detection interval and threshold is not a viable approach.

Therefore, since both gradual and sudden network condition changes are expected to happen in dynamic and heterogeneous networks, e.g., 5G mmWave [42], we leverage the drift theory [48] to observe the variability of network conditions. In particular, FALCON adopts the well-known Bayesian change point detection algorithm [49] for monitoring loss rate and RTT on the available paths. On the one hand, the RTT is a continuous signal, and thus it can be used as is in the Bayesian change point detection algorithm; on the other hand, packet loss is a binary information (i.e., a packet can be either lost or not lost). To tackle this aspect, FALCON counts the packet losses over groups of packets, thus moving from a Bernoulli distribution to a binomial distribution of packet losses, and finally obtaining a relatively continuous signal.

As shown in Algorithm 1, upon detection of a change in network conditions, the online learning module selects the meta-model in $\boldsymbol{\Theta}_{\mathsf{S}}$ that covers the range that the current network condition belongs to by checking the network conditions against $\boldsymbol{R}_{\mathsf{S}}$ over a short period of data transmission. When localizing the current network condition, there might exist bias due to the noise. Recall that the meta-model covers a range of link characteristics in FALCON, which reasonably tolerates these biases. Once the meta-model is selected, FALCON performs a K-step fine-tuning of the meta-model and, via DQN, derives the scheduling policy to adopt. Finally, FALCON deploys and uses the new policy until a new change is detected. Since the learning agent does not grow its knowledge base from null rather from the shared knowledge, the cost of adaptation is fairly small as shown in Section VI.

B. Learning Elements

As anticipated in Section IV-A, FALCON uses a DQN architecture for deriving the policy at run-time, and exploits the meta-learning paradigm to speed up such derivation while preserving accuracy. Therefore, the entire framework is a MDP that FALCON solves via *meta-learning plus DQN*. In the following, we provide a few more details on the learning elements of the overall framework.

State space: The state in a MDP is the information observed by a learning agent on the status of the environment that the agent is facing during the learning process. In our scenario, FALCON is the learning agent and the environment state is defined via transport layer parameters of the available paths, i.e., CWND, number of Inflight Packets (InP), Send Window (SWND) and RTT. The first three features are normalized by RTT to impose a tight connection to the throughput, that is the reward FALCON obtains while operating, as introduced later.

Action space: This set includes the actions FALCON can select when it deploys a scheduling policy, and upon which it gets a reward. In our scenario, the available actions depend on the number of available paths. In this work, we mostly consider two available paths, which is a common assumption in 5G multi-connectivity scenarios [5]. However, the action set of FALCON can be naturally extended and include more paths. Hence, by taking an action, FALCON decides about the path to use for exchanging a data packet. In the context of multipath scheduling, this indicates that the action set can be different when a path is congested or not congested, which complicates the learning agent. We choose to naturally inherit this information from the state while appointing the path in a straightforward manner regardless of the path's current congested status.

Reward function: As common in a MDP, FALCON aims at maximizing a so-called discounted return, where the instantaneous rewards, corresponding to the throughput obtained upon selection of a path, are cumulated after being discounted via a so-called discount factor, that can be interpreted as the interest of the scheduler in maximizing short vs. long-term return. The employment of the discounting factor ensures that the impact of the current action decreases over time.

RL algorithm: As anticipated above, FALCON uses DQN in the online learning module for deriving the scheduling policy. DQN is a well-known model-free algorithm that does not require any knowledge of the state transition probability distribution and the reward function. On the contrary, it just needs to observe the instantaneous rewards obtained when actions are selected, and the corresponding transition across states. While considering complexity as a primary factor, we select DQN also due to its popularity, which makes possible a direct comparison with other DQN-based state-of-the-art schedulers, as introduced in Section II-B. However, it is worth to mention that FALCON is based on a rather flexible framework and can be thus easily extended toward the adoption of other algorithms in the online learning module.

Exploration vs. exploitation: Due to the presence of the information exchange between the offline and online learning modules, FALCON requires a certain degree of balance between exploration and exploitation. Hence, it adopts a fixed ϵ-greedy exploration mechanism, with ϵ not decaying over time. In particular, a relatively large value of ϵ, denoted ϵ_l, is used when setting up these initial meta-models, aiming for higher sampling efficiency; on the contrary, a relatively small value of ϵ, that is, ϵ_s, is used when the meta-models are continually updated and also when the selected meta-model is fine-tuned to derive the scheduling policy.

Synchronous vs. asynchronous learning: In the original proposal of DQN and common deep RL paradigms, the interaction with the environment and the update of the neural network happen in a synchronous fashion. However, these synchronous operations do not work well in a real system where there usually exist either soft or hard real-time requirements. For example, in our case, the online update of the neural network could block the scheduling routine in the communication stack. We thus employ asynchronous updates [50], implemented by using a separate process for the online learning: A network process is in charge of the data collection and performs the scheduling while a trainer process is in charge of neural network updates based on the collected data.

V. Experimental setup

In this section we present the experimental setup, including the configuration of FALCON, the selected baseline multipath scheduling algorithms, and the experimental environment.

A. Configuration of FALCON

We implement the learning components of FALCON based on `keras-rl` [51], a popular deep reinforcement learning library. We employ a fully connected neural network with three hidden layers and a rectified linear activation function (ReLU) as the activation function. The learning rate of the neural network is 0.001, while ϵ_l and ϵ_s are 0.3 and 0.1, respectively. The mini-batch size is 32 and K is 16. For the ranges of network conditions that the meta-models cover over each path, we implement that the packet loss rate can be in between $[0, 1)\%$, $[1, 5)\%$, and $[5, 100]\%$; mean RTT can be in between $[0, 50)$ ms, $[50, 200)$ ms, and $[200, +\infty)$ ms; the ratio of RTT deviation to mean RTT can be in between $[0, 40)\%$, $[40, 80)\%$, and $[80, +\infty)\%$. Thus, one path, by combination, can have 27 different coarse-grained states and two paths, by combination, can have 729 different coarse-grained states. Accordingly, the number of meta-models in total is 729. The online experiences are periodically written to a comma-separated values (CSV) file and the neural networks representing the meta-models are saved into the Hierarchical Data Format version 5 (HDF5) file.

We set the specific parameters of FALCON based on our experimental analyses in Section VIII-A. We believe these are reasonable design choices in practice, and note that our analysis enables tuning this parameter to accommodate other scenarios. Unless stated otherwise, our evaluation of FALCON uses these default values.

B. Configurations of the protocol stack

At the transport layer, we perform our analysis using MPQUIC due to the increasing interest in QUIC-based applications. Accordingly, the QUIC is originally implemented within `quic-go` [52] and, based on which, one of the earliest versions of MPQUIC is implemented and adopted in this work. Further we use the default multipath congestion control algorithm in the adopted MPQUIC code base, i.e., Opportunistic Linked-Increases Algorithm (OLIA) [53].

At the application layer, we perform both bulk transfer and web download to evaluate the aggregation capability of the multipath schedulers. For the bulk transfer, each experiment run performs an `HTTP GET` request for a file of 2 MB, and records the download times. For the web download, we consider web pages from different websites including Google, Github, and Stackoverflow, as shown in Table II, and we record the download times. Transport layer state variables are reset before each request. To ensure statistically significant results, for each path configuration, we repeat the experiments 120 times for each multipath scheduler.

C. Benchmark Algorithms

We select **minRTT** and **BLEST** as the representative algorithms of the schedulers based on predefined rules, for two

Table I
EMULATION PARAMETERS FOR THE STATIC SCENARIO, COLLECTED FROM MEASUREMENTS IN THE LITERATURE.

Parameter	Link Technology		
	5G [42]	*4G* [42]	*WLAN* [54][55]
Bandwidth [Mbps]	$\sim$ 1100	140	30
Round trip time (RTT) [ms]	27.4 $\pm$ 6.4	29.2 $\pm$ 4.8	20.0 $\pm$ 10.0
Packet Loss Rate [%]	0.1	0.1	0.7

Table II
PARAMETERS OF SELECTED WEBSITES FOR THE WEB DOWNLOAD TEST.

Website	Parameters	
	Number of Objects	*Size*
Google	8	1.3 MB
Github	61	4.5 MB
Stackoverflow	120	7 MB

reasons: 1) the strategies they exploit consider the challenges originating not only from the homogeneous networks but also the heterogeneous networks; 2) recent evaluations show that they either perform similarly or better than other schedulers belonging to the same category (e.g., RR and ECF) [6], [9].

For the offline learning-based schedulers, as there is not any multipath scheduling algorithms in the literature, we refer to a recent implementation in the area of ABR streaming [24], and implement a DQN-based multipath scheduling algorithm, named **DQN-Off**. In the implementation, while keeping the concept of offline training, we utilize the off-policy characteristics of DQN rather than a simulated environment to achieve this goal. That means, DQN off-policy is able to learn from information retrieved from past experience rather than direct interaction with the environment. Moreover, we use the same learning elements used by FALCON, described in Section IV-B.

For Type-I online learning based schedulers, we refer to the **M-Peekaboo** algorithm that can exploit the linUCB and stochastic adjustment algorithm to learn the scheduling policy [6]. For Type-II online learning based schedulers, the DQN-based scheduler designed in [8] fails to provide clear performance gains over schedulers based on predefined rules, while [7] shows performance gains. However, the authors of [7] do not disclose the source code, and we do not have enough information to reproduce the work. Hence, we use all the information that can be extracted from [8] and [7], and implement a DQN-based online multipath scheduler, i.e., **DQN-On**. In particular, within the framework provided by [8], we exploit the same state space, action space, and reward function used by FALCON, to have a higher granularity representation in the constructed MDP. Then, we also adopt the asynchronous online update mechanism used in [7], originally proposed in [50], to speed up the training time of DQN when applied to real-world applications. Although [7] refers to this asynchronous online update mechanism as a combination of online and offline learning, we highlight that this is in essence an implementation choice of inter-process communication, and thus it differs from the concept of online and offline learning defined in this work.

D. Experimental environment

We perform experiments over both emulated and real-world *urban canyon* environments. In both cases, we consider two scenarios: *static*, where we assume the user is stationary, and *mobile*, where the user is walking and/or driving a vehicle.

In the emulated experiments, aiming at a controlled but realistic evaluation, we leverage link characteristics derived from real measurements, with both network traces and statistical values, as shown in Table II. The environment is emulated using `Mininet` [56]. Regarding path characteristics (i.e., bandwidth, latency, and packet loss), we use values measured against a content-server close to the radio infrastructure, mimicking 5G edge deployments. In the static scenario, we showcase multipath transport between 4G and 5G paths and between 4G and WLAN paths as well as between 5G and WLAN paths. Our motivation to evaluate all these options is due to the proposed ATSSS architecture from 3GPP. In the mobile scenario, we showcase multipath transport over two 5G networks in a driving scenario [42].

In the real-world experiments, we analyze multipath transport between 5G and WLAN in the static scenario. In the mobile scenario, we showcase multipath transport over 4G and 5G in a driving test.

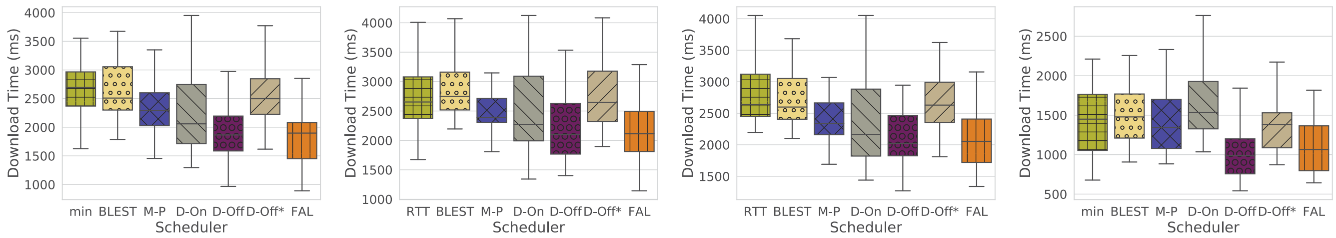
VI. Emulated Experiments

In this section, we compare the performance of FALCON with state-of-the-art multipath schedulers in a wide range of emulated experiments. We use the bulk transfer case to analyze the performance of the multipath schedulers in static and mobile scenarios (Section VI-A), and provide more insights on the behavior of FALCON and other schedulers in terms of how quickly they adapt to time-varying network conditions (Section VI-B). Finally, we further validate the robustness of FALCON in web download scenarios (Section VI-C).

A. Performance in Static and Mobile Scenarios

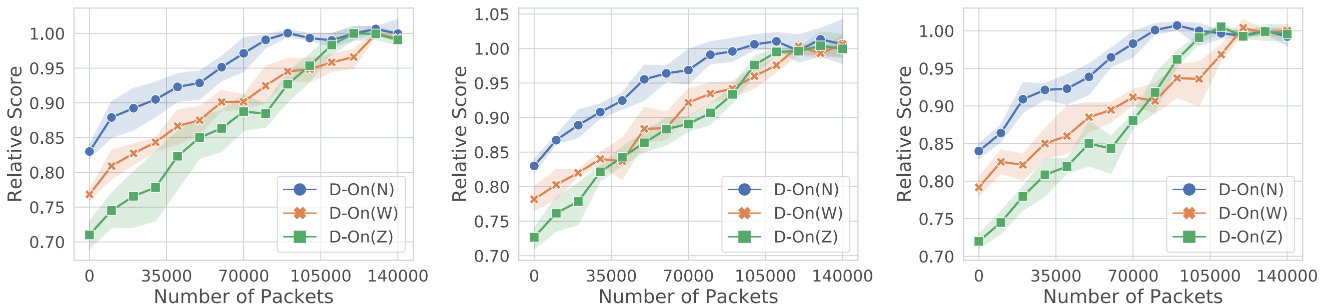
We first evaluate the performance of different multipath schedulers in static and mobile scenarios. We focus on the analysis of schedulers based on learning while keeping schedulers based on predefined rules as a reference. For the schedulers based on learning with online approach (FALCON, DQN-On, M-Peekaboo), we assume: (i) they were not trained over the examined network conditions beforehand, and (ii) they have no buffered online data at the beginning of each experiment. On the other hand, to directly compare the impact of an approach with offline pre-knowledge, we assume that DQN-Off was trained over the examined network conditions beforehand.

Figure 2 presents the performance of different schedulers under different scenarios as described in Section V-D. For the static case (Figure 2 a-c), we observe that all the schedulers based on learning (FALCON, DQN-On, DQN-Off, M-Peekaboo) outperform the schedulers based on predefined rules (minRTT, BLEST) with up to 34.5% shorter median download time. Concerning schedulers based on learning, we observe that schedulers utilizing deep learning, including FALCON, DQN-Off, and DQN-On, outperform M-Peekaboo



(a) Static: 4G and 5G (b) Static: 4G and WLAN (c) Static: 5G and WLAN (d) Driving: 5G and 5G

Figure 2. Performance of FALCON and other multipath schedulers in the static and mobile network conditions over 4G, 5G, and WLAN.



(a) 4G and 5G (b) 4G and WLAN (c) 5G and WLAN

Figure 3. Convergence test of DQN-On in different configurations over 4G, 5G, and WLAN.

with up to 19.3% shorter median download time. As all the schedulers adapt to the presented static network condition, this indicates that applying a model of high complexity is beneficial for improving the adaptation accuracy. We also observe that the performance of FALCON is similar to DQN-Off and clearly better than DQN-On. This indicates that FALCON can adapt faster compared to DQN-On thanks to its few-shot online learning, which allows to achieve the same accuracy as DQN-Off.

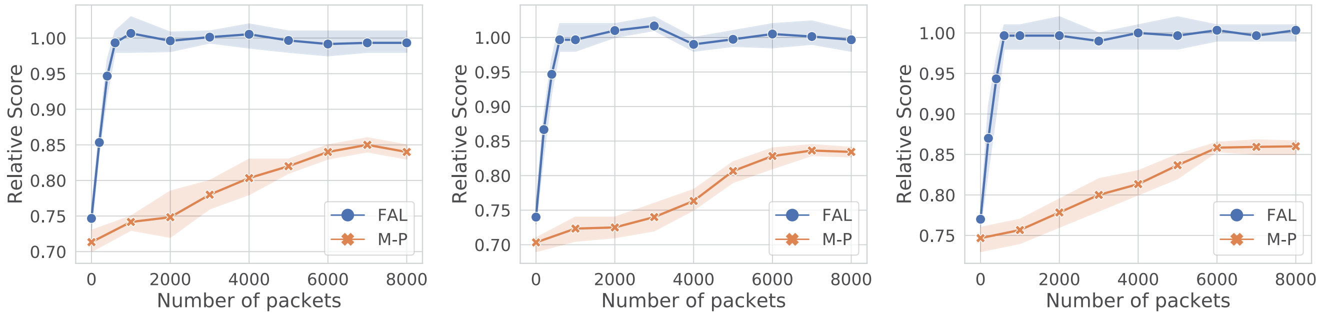
Note that we assumed DQN-Off was trained over the examined network conditions beforehand and is able to deploy an accurate model without the additional cost of online learning. However, it is rarely the case that the online data is fully and well-aligned with the offline data under realistic settings. To capture this effect, we consider the scenario, where the model obtained during training deviates from the current network conditions by only a 5% decrease in terms of the RTT variation and loss rate of the paths. We denote DQN-Off under these new deviated network conditions as DQN-Off*. We observe that there is a significant performance drop of DQN-Off* compared to DQN-Off with up to 34.5% longer median download time. Its performance is similar to the schedulers with pre-defined rules. This indicates that DQN-Off lacks the ability to adapt, hence negatively impacting its practicality under realistic settings.

Next, we evaluate the performance of FALCON and baseline schedulers in a mobile scenario. We illustrate the performance of different schedulers in the trace-driven mobile network conditions in Figure 2(d). We observe that the performance gain of M-Peekaboo over the schedulers based on predefined rules decreases compared to that of the static case, since it does not adapt fast enough to the less predictable changes of network conditions. M-Peekaboo, however, outperforms DQN-On by an 18.9% shorter median download time, since it has a more lightweight learning mechanism and, thus, a shorter adaptation time. The adaptation time of DQN-On is quite long because of the intrinsic slow convergence time of DQN, which we investigate separately in Section VI-B1. Thanks to the few-shot online learning, FALCON still clearly outperforms the online learning based schedulers, reaching a 16% shorter median download time compared to M-Peekaboo. DQN-Off performs slightly better than FALCON as it does not have the cost of few-shot learning during frequent network condition changes. However, when we introduce a model deviation, as done for the static scenario, we observe again that DQN-Off* performance drops significantly, since it lacks the ability to adapt to the deviated network conditions.

B. *A Closer Look into Adaptation Time*

We now examine in depth the factors that impact the adaptation time for the schedulers based on learning with online adaptation, i.e, DQN-On, M-Peekaboo, and FALCON.

1) Convergence Test: We perform a convergence test to explore the convergence behavior of DQN-On, M-Peekaboo, and FALCON. We define the relative score as the ratio between the median file download time obtained by DQN-Off and the median file download time obtained by the scheduler under test. We use the relative score to illustrate how online learning algorithms evolve over time, thus, we evaluate this score as a function of the online learning cost, i.e., the number of online packets at the transport layer. For each scheduler, we perform the test 10 times. For DQN-On, we also consider the impact of buffered online data due to previous training.



(a) 4G and 5G (b) 4G and WLAN (c) 5G and WLAN

Figure 4. Convergence test of FALCON and M-Peekaboo over 4G, 5G, and WLAN.

Therefore, we not only investigate the convergence of DQN-On with zero buffered online data, i.e., no previous training, (denoted by DQN-On(Z)) but also with a narrow vs. wide range of buffered online data (denoted by DQN-On(N) and DQN-On(W), respectively). In particular, DQN-On(N) was trained beforehand over two network conditions that have, compared to current network conditions, a 3% decrease or a 3% increase of RTT variation and loss rate on the available paths. DQN-On(W) was instead trained over four network conditions having 3% and 6% decrease and increase of the same indicators, respectively. The experience on the network conditions is obtained by exploiting a learning budget of 100 packets exchanged under each of these conditions.

Figure 3 shows the results of the convergence test for DQN-On with different amount of buffered online data. The number of packets for reaching convergence is very different when we compare DQN-On and FALCON / M-Peekaboo; therefore we show the results of DQN-On in Figure 3 and the results of FALCON / M-Peekaboo in Figure 4. We observe that DQN-On requires a large amount of data to converge, in the order of 100,000 packets. DQN-On(N) has a relatively higher score at the beginning and shows earlier convergence compared to DQN-On(W). This is due to two main reasons: first, the network conditions it has trained over have a higher similarity to the current network conditions; second, the total number of packets in its learning budget is smaller (200 for DQN-On(N) vs. 400 for DQN-On(W)), so in DQN-On(N) the online data from current network conditions dominates faster the buffered online data, ultimately speeding up adaptation. Similarly, DQN-On(W) has a relatively higher score than DQN-On(Z) at the beginning, due to the training beforehand, but DQN-On(Z) converges earlier than DQN-On(W), since it does not need to nullify the impact of buffered online data deviating from current network conditions. The analysis indicates that buffered online data that deviates from current conditions may harm convergence. Note that we allocated a small amount of online data that are relatively close to the current network conditions (deviations are within 6%); In more realistic settings, even more data with wider deviated ranges could be buffered, leading to a continuous slow down of the adaptation time of DQN-On.

Next, we illustrate the results of the convergence test for M-Peekaboo and FALCON in Figure 4. We observe that

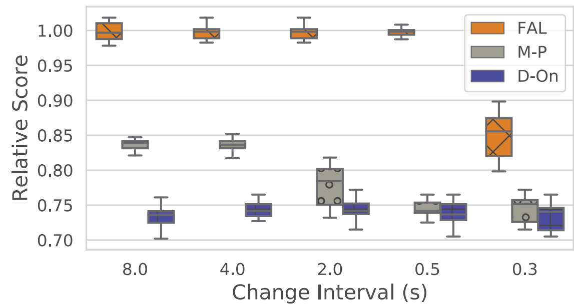


Figure 5. Stress test of FALCON, M-Peekaboo, and DQN-On.

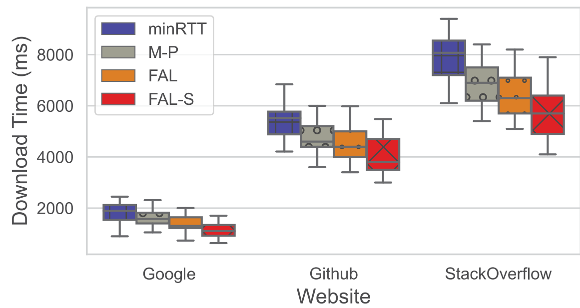


Figure 6. Web download test of FALCON, M-Peekaboo, and minRTT.

FALCON and M-Peekaboo achieve convergence with an approximate learning cost of 600 and 6,000 packets, respectively. These values are much smaller than that of DQN-On. Meanwhile, the paradigms of FALCON and M-Peekaboo are free of the impact of the buffered online data. Further, we observe that M-Peekaboo has a relatively fast convergence speed but its relative score is lower than FALCON, due to its simpler learning model. On the other hand, by combining offline and online learning, FALCON not only converges faster but also achieves a higher score compared to M-Peekaboo.

2) Stress Test: We also perform a stress test to examine how fast and accurately FALCON, DQN-On, and M-Peekaboo adapt to changing network conditions. In order to isolate the impact of adaptation, none of the schedulers has buffered online data beforehand. We define a change interval and, under each change interval, we generate 24 different network conditions where the characteristics of each path are randomly generated in the range formed by the minimum and maximum of the characteristics shown in Table II. At the end of each change interval, we calculate the relative score of the multipath scheduler using the approach presented in Section VI-B1.

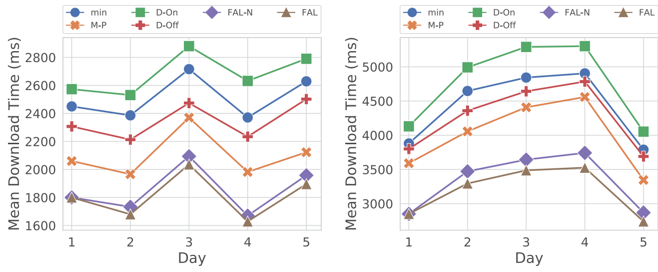
Figure 5 shows the relative score of each multipath scheduler under the stress test with change intervals of 8.0, 4.0, 2.0, 0.5, and 0.3 seconds, respectively. We observe that DQN-On already struggles when the change interval is 8.0 seconds, as indicated by a relative score much less than 1. This is in line with the results on the convergence behavior of DQN-On in Section VI-B1. We also observe that M-Peekaboo struggles with a change interval of 2 seconds, as indicated by the drop of its relative score compared to the scores obtained with change intervals of 8.0 and 4.0 seconds. We further observe that FALCON performs very well up to a change interval of 0.5 seconds. Then, it experiences a performance drop when the change interval is equal to 0.3 seconds. When both FALCON and M-Peekaboo can catch up with the change of network conditions (e.g., the change interval of 4 seconds), FALCON reaches a higher performance than M-Peekaboo. In all cases, FALCON shows higher scores compared to all the other schedulers, ultimately highlighting a significantly higher both adaptation accuracy and speed.

C. Multi-streaming support for Web Services

In Sections VI-A and VI-B, we have examined the effectiveness of FALCON for bulk transfer services. In this section, we will examine the extensibility of FALCON for web services.

For the web experiments, we use the stream multiplexing feature of MPQUIC, an important feature that is planned to be exploited in HTTP/3. Therefore, we follow the approaches in the existing literature for dealing with stream multiplexing, and utilize a weighted round robin stream scheduling approach to download webpage objects based on their position in the dependency tree of the webpage [57], [58], [59]. Moreover, considering the multi-streaming feature of MPQUIC, we plug in partially different contents within the framework of FALCON for web download from that for bulk transfer. Although both are subject to the same algorithm (i.e. single streaming is a special case of multi-streaming), we denote the one with multi-streaming support as FALCON-S and FALCON for the one with single-streaming support, just to ease the presentation in the experiment. There are two main differences between FALCON and FALCON-S: 1) FALCON-S takes the send windows of each object stream as the state information while FALCON treats the send window as a whole for the state information; 2) FALCON-S splits the congestion window based on the weights of concurrent streams as the state information for each stream while FALCON treats the congestion window as a whole for the state information.

We perform the web experiment within the mobile scenario as defined in Section VI-A, to better illustrate the algorithm's adaptation ability. Figure 6 shows the download time of minRTT, M-Peekaboo, FALCON, and FALCON-S for different web pages. We observe that FALCON still has a clear performance gain over the other multipath schedulers. Furthermore,



(a) Static: 5G and WLAN (b) Driving: 4G and 5G

Figure 7. Performance of FALCON and other multipath schedulers in real-world static and mobile network conditions over 4G, 5G, and WLAN.

FALCON-S outperforms FALCON, reaching up to 13.6% shorter download time. The results shows that it is possible to use FALCON across different applications, indicating the robustness of FALCON. Moreover, simple application-specific tuning can be applied to FALCON, in order to customize it for specific applications, eventually indicating the flexibility of FALCON.

VII. Real-world Experiments

We now present the evaluation of the schedulers in real-world experiments in both static and mobile scenarios.

The static scenario is set up over 5G from a network provider and WLAN, while the mobile scenario is set up for a vehicle moving at a nearly constant speed of 30 km/h, over 5G from the same network provider and 4G from a different network provider. The evaluation is carried out in the afternoon, while the data for creating the meta-models in FALCON and for training DQN-Off is collected 5 days before the evaluation, in the morning. We perform the evaluation over 5 consecutive days and, at the end of each day, FALCON performs the offline update of the meta-models. To illustrate the effect of the offline update, we also show the performance of FALCON with no offline update, denoted FALCON-N (in these settings, FALCON and FALCON-N are identical in the first day of evaluation).

Figure 7 illustrates the performance of FALCON and other multipath schedulers in real-world network conditions. We note that in both static and mobile scenarios, the performance of DQN-On is always lower compared to all schedulers, due to the frequent retraining as the network conditions change. While in the emulated environment, DQN-On can converge (see Section VI-B1), in the real-world environment, even in the static scenario, the state transitions are more frequent due to the dynamicity of real networks. DQN-Off performs consistently better than DQN-On and minRTT, but worse than M-Peekaboo, FALCON, and FALCON-N, as it lacks the ability to adapt online. Restricted by its adaptation time, M-Peekaboo has worse performance in the mobile scenario than in the static case. FALCON and FALCON-N outperform M-Peekaboo with up to 23.6% and 18.7% shorter mean download time, respectively. In particular, FALCON outperforms FALCON-N, with a gain indicating that the effect of updating the meta-models is incremental. To summarize, the results

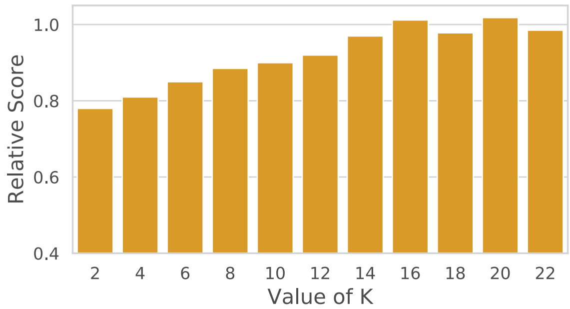


Figure 8. The impact of K on FALCON's performance.

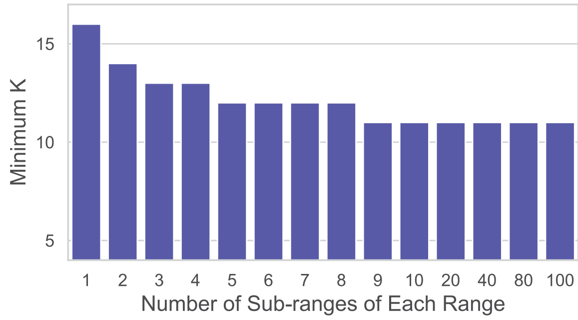


Figure 9. The impact of the number of sub-ranges (also, the number of meta-models) on the optimal K.

show a rather high generalizability of the meta-models learned over the distribution of network conditions.

VIII. FALCON's Configuration Parameters and Overhead

In this section, we study the impact of FALCON's configuration on the obtained performance (Section VIII-A). We then discuss the overhead of FALCON (Section VIII-B).

A. A Study into FALCON's Configuration

We study the impact of the configuration parameters adopted for FALCON on the observed performance, e.g. adaptation speed and accuracy. More specifically, we address the selection of K and the number of meta-models, which are directly related to FALCON operations.

1) Selection of K: We first study the selection of K and its impact on adaptation speed and accuracy. Recall that K is the number of online training steps to be performed for fine-tuning the pre-built meta-models. We expect that for any given network condition, the online model should converge within K steps. Since we perform a mini-batch gradient descent with the size of 32 for each step, the learning overhead in terms of the number of packets becomes the number of steps multiplied by the mini-batch size. Thus, we seek for the smallest value of K that guarantees fast and accurate adaptation.

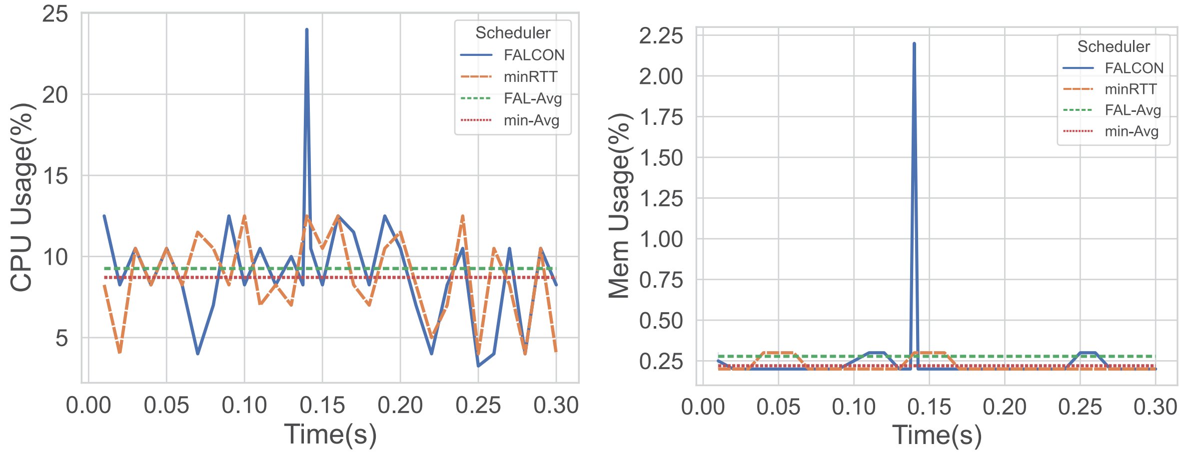
Figure 8 shows the performance of FALCON in terms of relative score as a function of K. We first observe that, when K is relatively small, FALCON does not show significant gains. This is because the meta-learning mechanism is struggling to find meta-models that can converge within the K steps. The performance saturates when $K = 16$, which is therefore selected as the parameter adopted in FALCON.

2) Number of Meta-models: Next, we study the impact of the number of meta-models we employ. Recall that, for a combination of our defined range of link characteristics, we train one meta-model to bootstrap. To obtain a higher amount of meta-models over the defined range of link characteristics, we divide each range into multiple sub-ranges (e.g., divide the $[0, 1)\%$ range of loss rate into a number of sub-ranges) and train one meta-model for each combination of sub-ranges.

Figure 9 shows the minimum value of K (as analyzed in Section VIII-A1) as a function of the number of sub-ranges for each range of link characteristics (the original one is 1). We observe that the value of K slowly decreases as we increase the number of sub-ranges for each range. However, even if the number is set to a relatively large number (e.g., 100), the minimum K is still relatively large. Indeed, the meta-model still requires a certain number of training steps before converging to the optimal values.

In theory, when the number of sub-ranges (also, the number of meta-models) is sufficiently high, i.e., offline and online scenarios will converge, the minimum value of K will be zero, meaning that there will be no need for online adaptation. Nevertheless, this is not practical for the reasons we present in Section III-A. Furthermore, FALCON requires the estimation of current network conditions, in order to map such conditions to one of the pre-built meta-models. The estimation error can easily cause disturbance in selecting the meta-models if the number of meta-models is too large. For this reason, we keep the original number of meta-models for FALCON that is practical and avoids the estimation errors with the satisfactory performance and adaptation speed.

3) Discussion on Hyperparameter Selection: Learning based systems cannot avoid the necessity to employ hyperparameters in their algorithms. The search for the hyperparameters to use is an optimization problem, which is often solved heuristically in a trial-and-error manner. In extreme cases, the trial-and-error process can be automated, and this is known as automated machine learning [60]. In all cases, the higher the complexity of the model and the task, the longer the time per trial would be. Thus, this approach is normally used with small-sized models and datasets so that the iterations for optimization can be completed until a set of parameters is found. Since FALCON is set up over a significant size of models and data, this optimization approach is not feasible for FALCON, just as for most of the other practical machine learning systems that eventually employ intuitive hyperparameters with human-in-the-loop manual optimization (tuning). Therefore, as regards to DQN-related parameters, we adopt the common parameters, since they already bring significant gains for FALCON against other schedulers. We further observe that the selection of these parameters is subject to machine learning engineering aspects, and their optimization may result in further improvements.



(a) CPU Cost

(b) Memory Cost

Figure 10. Overhead of FALCON and minRTT in the perspectives of CPU usage percentage and memory usage percentage.

B. System Overhead of FALCON

A deep learning system normally consists of a training phase and an inference phase (i.e., the interpretation of the neural network). For FALCON, the training phase is partially completed offline and partially completed online along with the inference phase; the inference phase is completed online. The offline training phase is of higher system overhead but free of impact on the deployment of FALCON, since it occurs in an offline manner.

We perform an experimental analysis to investigate the system overhead of FALCON in the online phase. We record the central processing unit (CPU) usage and memory usage of FALCON and minRTT over a 0.3 seconds period with a single change of network conditions. The CPU of the server is Intel-i5 (2.50 GHz) of two cores and the size of the memory of the server is 8 GB. Figure 10 shows both the real time usage and the average usage of FALCON and minRTT. First, we observe that the average CPU usage of FALCON is only 3% higher than minRTT and the memory usage of FALCON is on average only 6% higher than minRTT. Recalling that we have a change of network conditions within the analyzed time frame of 0.3 seconds, i.e., the worst-case scenario where FALCON can be used, the CPU and memory usage gaps between FALCON and minRTT are likely to be even smaller when the network conditions change less frequently. Thus, overall FALCON does not bring significant system overhead. For the real time usage, we do observe a significant spike in terms of CPU and memory usage. This spike happens when FALCON performs the online training with the gradient calculation. However, we do not observe any extra system overhead in the online inference phase, because FALCON utilizes a neural network model with a relatively simple architecture and thus with low computational complexity. Lastly, FALCON is deployed in the server in the context of this paper, thus the extra CPU and memory costs are not as significant issues as they would be on the client device.

However, we do not limit the applicable scenarios of FALCON to server side deployment. A path selection on the client side can also employ FALCON. In such a context, the client would be a mobile device, constrained by power consumption. For the online inference side, as mentioned above, FALCON and other schedulers should hold similar power consumption, as inferred from the similar CPU and memory utilization. Considering a scenario where FALCON might employ a Neural Network (NN) model of higher complexity, and thus larger inference overhead, embedded software and hardware solutions such as ARM Common Microcontroller Software Interface Standard (CMSIS) NN software library [61], Field-Programmable Gate Array (FPGA), and Graphics Processing Unit (GPU), can make the inference more efficient.

IX. Discussion and Conclusion

Learning-based networking systems have received much attention of late, as well intriguing the field of multipath scheduling. However, the deployment of existing learning-based multipath schedulers fails to be functional in the aspects of achieving a fast and accurate adaptation.

In this paper, we propose FALCON, a learning-based multipath scheduler that can adapt fast and accurately to time-varying network conditions by combining the benefits of online and offline learning. Through extensive emulations, we show that FALCON is able to consistently outperform all state-of-the-art schedulers by adapting to the network conditions in a fast and accurate manner. Our real-world experiments confirm that FALCON performs well also under realistic network settings.

We see two main future directions for this work. Firstly, in this paper, we have demonstrated the possibility of applying DQN within FALCON, but we will also consider applying other deep learning approaches to enhance the performance of FALCON. Secondly, we plan to interpret and understand the learning outcome of FALCON (i.e., in the form of NN) to potentially deduce the guaranteed performance bound.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[2] M.-T. Suer, C. Thein, H. Tchouankem, and L. Wolf, "Multi-connectivity as an enabler for reliable low latency communications—an overview," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 156–169, 2019.

[3] A. Ford *et al.*, "TCP Extensions for Multipath Operation with Multiple Addresses," Internet Requests for Comments, RFC Editor, RFC 8684, March 2020, Accessed: Dec. 2020. [Online]. Available: http://www.rfc-editor.org/rfc/rfc8684.txt

[4] Q. De Coninck and O. Bonaventure, "Multipath QUIC: Design and Evaluation," in *Proc. of ACM Int'l Conf. on Emerging Networking Experiments and Technologies (CoNEXT'17)*. ACM, 2017.

[5] *23.501:System Architecture for the 5G System*, 3GPP, 03 2020, v16.4.

[6] H. Wu, G. Caso, S. Ferlin, O. Alay, and A. Brunstrom, "Multipath Scheduling for 5G Networks: Evaluation and Outlook," in *IEEE Communications Magazine*. IEEE, 2021.

[7] H. Zhang, W. Li, S. Gao, X. Wang, and B. Ye, "Reles: A neural adaptive multipath scheduler based on deep reinforcement learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1648–1656.

[8] M. M. Roselló, "Multi-path scheduling with deep reinforcement learning," in *2019 European Conference on Networks and Communications (EuCNC)*. IEEE, 2019, pp. 400–405.

[9] H. Wu, Ö. 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, 2020.

[10] H. Wu, G. Caso, S. Ferlin, Ö. Alay, and A. Brunstrom, "Multipath scheduling for 5g networks: Evaluation and outlook," *IEEE Communications Magazine*, vol. 59, no. 4, pp. 44–50, 2021.

[11] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[12] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[13] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kuehlewind, "Multipath extension for quic," Working Draft, IETF, Internet-Draft draft-lmbdhk-quic-multipath-00, October 2021. [Online]. Available: https://datatracker.ietf.org/doc/draft-lmbdhk-quic-multipath/00/

[14] S. Ferlin *et al.*, "BLEST: Blocking Estimation-based MPTCP Scheduler for Heterogeneous Networks," in *Proc. of Int'l Federation for Information Processing Networking Conference (IFIP Networking'16)*, 2016.

[15] Y.-s. Lim *et al.*, "ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths," in *Proc. of ACM Int'l Conf. on Emerging Networking Experiments and Technologies (CoNEXT'17)*. ACM, 2017.

[16] A. Frommgen, T. Erbshäußer, A. Buchmann, T. Zimmermann, and K. Wehrle, "ReMP TCP: Low latency multipath TCP," in *IEEE ICC*, 2016.

[17] H. Lee, J. Flinn, and B. Tonshal, "Raven: Improving interactive latency for the connected car," in *ACM MobiCom*, 2018.

[18] Y. E. Guo, A. Nikravesh, Z. M. Mao, F. Qian, and S. Sen, "Accelerating multipath transport through balanced subflow completion," in *ACM MobiCom*, 2017.

[19] H. Shi, Y. Cui, X. Wang, Y. Hu, M. Dai, F. Wang, and K. Zheng, "STMS: Improving MPTCP throughput under heterogeneous networks," in *USENIX ATC*, 2018.

[20] E. Dong, M. Xu, X. Fu, and Y. Cao, "A loss aware MPTCP scheduler for highly lossy networks," *Computer Networks*, 2019.

[21] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, O. Alay, and N. Kuhn, "Low-latency scheduling in MPTCP," *IEEE/ACM Transactions on Networking (ToN)*, vol. 27, no. 1, pp. 302–315, 2019.

[22] S. K. Saha, S. Aggarwal, R. Pathak, D. Koutsonikolas, and J. Widmer, "Musher: An agile multipath-tcp scheduler for dual-band 802.11 ad/ac wireless lans," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[23] K. Winstein and H. Balakrishnan, "Tcp ex machina: Computer-generated congestion control," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 123–134, 2013.

[24] Z. Akhtar, Y. S. Nam, R. Govindan, S. Rao, J. Chen, E. Katz-Bassett, B. Ribeiro, J. Zhan, and H. Zhang, "Oboe: auto-tuning video abr algorithms to network conditions," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 44–58.

[25] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 197–210.

[26] H. Mao, M. Schwarzkopf, S. B. Venkatakrishnan, Z. Meng, and M. Alizadeh, "Learning scheduling algorithms for data processing clusters," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 270–288.

[27] J. Ren, X. Wang, J. Fang, Y. Feng, D. Zhu, Z. Luo, J. Zheng, and Z. Wang, "Proteus: network-aware web browsing on heterogeneous mobile systems," in *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, 2018, pp. 379–392.

[28] A. Padmanabha Iyer, L. Erran Li, M. Chowdhury, and I. Stoica, "Mitigating the latency-accuracy trade-off in mobile data analytics systems," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 513–528.

[29] M. Dong, T. Meng, D. Zarchy, E. Arslan, Y. Gilad, B. Godfrey, and M. Schapira, "{PCC} vivace: Online-learning congestion control," in *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*, 2018, pp. 343–356.

[30] J. Jiang, S. Sun, V. Sekar, and H. Zhang, "Pytheas: Enabling data-driven quality of experience optimization using group-based exploration-exploitation," in *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, 2017, pp. 393–406.

[31] T. Gilad, N. Rozen-Schiff, P. B. Godfrey, C. Raiciu, and M. Schapira, "Mpcc: online learning multipath transport," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, 2020, pp. 121–135.

[32] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, and S. Lu, "Smartcc: A reinforcement learning approach for multipath tcp congestion control in heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2621–2633, 2019.

[33] R. Kemker, M. McClure, A. Abitino, T. L. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[34] T. Huang, C. Zhou, X. Yao, R.-X. Zhang, C. Wu, B. Yu, and L. Sun, "Quality-aware neural adaptive video streaming with lifelong imitation learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2324–2342, 2020.

[35] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.

[36] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, 2016, pp. 3981–3989.

[37] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[38] J. Schmidhuber, "Optimal ordered problem solver," *Machine Learning*, vol. 54, no. 3, pp. 211–254, 2004.

[39] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, 2016, pp. 1842–1850.

[40] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *arXiv preprint arXiv:1611.05763*, 2016.

[41] T. Munkhdalai and H. Yu, "Meta networks," *Proceedings of machine learning research*, vol. 70, p. 2554, 2017.

[42] A. Narayanan *et al.*, "A First Look at Commercial 5G Performance on Smartphones," in *Proc. of The Web Conference*, 2020, pp. 894–905.

[43] P. Zhou, K. Cheng, X. Han, X. Fang, Y. Fang, R. He, Y. Long, and Y. Liu, "Ieee 802.11 ay-based mmwave wlans: Design challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1654–1681, 2018.

[44] J. Shi, M. Sha, and X. Peng, "Adapting wireless mesh network configuration from simulation to reality via deep learning based domain adaptation." in *NSDI*, 2021, pp. 887–901.

[45] N. H. Rotman, M. Schapira, and A. Tamar, "Online safety assurance for learning-augmented systems," in *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, 2020, pp. 88–95.

[46] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[47] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.

[48] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[49] R. P. Adams and D. J. MacKay, “Bayesian online changepoint detection,” *arXiv preprint arXiv:0710.3742*, 2007.
[50] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.
[51] M. Plappert, “keras-rl,” https://github.com/keras-rl/keras-rl, 2016.
[52] “Quic-go,” https://github.com/lucas-clemente/quic-go, 2019.
[53] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, “Mptcp is not pareto-optimal: Performance issues and a possible solution,” *IEEE/ACM Transactions On Networking*, vol. 21, no. 5, pp. 1651–1665, 2013.
[54] R. K. Sheshadri and D. Koutsonikolas, “On packet loss rates in modern 802.11 networks,” in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.
[55] C. Pei, Y. Zhao, G. Chen, R. Tang, Y. Meng, M. Ma, K. Ling, and D. Pei, “Wifi can be the weakest link of round trip network latency in the wild,” in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.
[56] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, “Reproducible network experiments using container-based emulation,” in *ACM CoNEXT*, 2012.
[57] J. Wang, Y. Gao, and C. Xu, “A multipath quic scheduler for mobile http/2,” in *Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019*, 2019, pp. 43–49.
[58] A. Rabitsch, P. Hurtig, and A. Brunstrom, “A stream-aware multipath quic scheduler for heterogeneous paths,” in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, 2018, pp. 29–35.
[59] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, “The quic transport protocol: Design and internet-scale deployment,” in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.
[60] Q. Yao, M. Wang, Y. Chen, W. Dai, Y.-F. Li, W.-W. Tu, Q. Yang, and Y. Yu, “Taking human out of learning applications: A survey on automated machine learning,” *arXiv preprint arXiv:1810.13306*, 2018.
[61] ARM, “Cmsis nn software library,” https://www.keil.com/pack/doc/CMSIS/NN/html/index.html, 2021.

Hongjia Wu is a Ph.D. candidate at Simula and OsloMet. He obtained his M.Sc. in Embedded Systems from TU Delft and B.Sc in Automatic Control from Northeastern University. His research interests include multipath protocols and robotic systems.

Özgü Alay Dr. Ozgu Alay received the B.S. and M.S. degrees in Electrical and Electronic Engineering from Middle East Technical University, Turkey, and Ph.D. degree in Electrical and Computer Engineering at Tandon School of Engineering at New York University. Currently, she is an Associate Professor in University of Oslo, Norway and Head of Department at Mobile Systems and Analytics (MOSAIC) of Simula Metropolitan, Norway. Her research interests lie in the areas of mobile broadband networks, multipath protocols and robust multimedia transmission over wireless networks. She is author of more than 70 peer-reviewed IEEE and ACM publications and she actively serves on technical boards of major conferences and journals.

Anna Brunstrom received a B.Sc. in Computer Science and Mathematics from Pepperdine University, CA, in 1991, and a M.Sc. and Ph.D. in Computer Science from College of William & Mary, VA, in 1993 and 1996, respectively. She joined the Department of Computer Science at Karlstad University, Sweden, in 1996, where she is currently a Full Professor and Research Manager for the Distributed Systems and Communications Research Group. Her research interests include Internet architectures and protocols, techniques for low latency Internet communication, multi-path communication and performance evaluation of mobile broadband systems including 5G. She has authored/coauthored over 170 international peer-reviewed journal and conference papers.

Giuseppe Caso is an Experienced Researcher at Ericsson Research (Radio Systems and Standards) in Kista, Sweden. In 2018-2021, he was a Postdoctoral Fellow with the MOSAIC Department at SimulaMet, Oslo, Norway. In 2016, he received the Ph.D. degree from Sapienza University of Rome, where he was a Postdoctoral Fellow until 2018. From 2012 to 2018, he has held visiting positions at Leibniz University of Hannover, King’s College London, Technical University of Berlin, and Karlstad University. His research interests include cognitive and distributed communications, resource allocation in cellular systems, IoT technology and evolution, and location-based services. He is an IEEE Member.

Simone Ferlin is a software researcher at Ericsson AB in radio networks. She received her Dipl.-Ing. degree in Information Technology with major in Telecommunications from Friedrich-Alexander Erlangen-Nuernberg University, Germany in 2010 and her PhD degree in computer science from the University of Oslo, Norway in 2017. Her interests lie in the intersection of cellular networks and the Internet, with her research focusing on computer networking, QoS and cross-layer design, transport protocols, congestion control, network performance, security, and measurements. Her dissertation focused on improving robustness in multipath transport for heterogeneous networks with MPTCP. She actively serves on technical boards of major conferences and journals in these areas.

POSTADRESSE:

OsloMet – storbyuniversitetet
Pilestredet 46
Postboks 4, St. Olavs Plass
0130 Oslo

OsloMet Avhandling 2022 nr 6
ISSN 2535-471X
ISBN 978-82-8364-377-0