POLITECNICO DI TORINO

Master's Degree in Embedded Systems

Master's Degree Thesis

Implementation and Evaluation of Multipath TCP (MPTCP) Schedulers For Reliable and Low-latency Car-to-Cloud Communication

Supervisors:
Dr. Riccardo CANTORO
Prof. Dr. -Ing. Christian WIETFELD
Co-Supervisor:
M.Sc. Johannes GÜLDENRING

Candidate:
Andrea NOTA

OCTOBER 2020

Contents

Abstract

Recent developments in the automotive and communication sectors enabled vehicles to operate as a highly mobile sensor platform. As the connected car concept continues to gain traction, Car-to-Cloud communication can support the transmission of collected data to the cloud, e.g., dynamic collection of new map information or localized weather forecasts. In addition to low latency and reliability, the data transmission requires a stable connection between a mobile vehicle and the cloud resources.

Compared with only a single Long Term Evolution (LTE) connection such as regular Transmission Control Protocol (TCP), leveraging multiple LTE interfaces (e.g. multipath protocols) can dramatically improve the transmission speed. However, Multipath Transmission Control Protocol (MPTCP) lacks the support to prioritize one path over another. In case the network providers have different costs, it may cause undesired network usage of the most expensive connection, resulting in a higher overall transmission cost.

The thesis focuses on the implementation in the Linux kernel and evaluation through MiniNet of Multipath Deadline and Preference Aware (MP-DPA), a multipath framework inspired to Multi Path Dynamic Adaptive Streaming over HTTP (MP-DASH) [1] with the awareness of network interface preferences. The basic idea behind MP-DPA is to strategically schedule packets' delivery while maintaining a reliable transmission and satisfying interface preferences.

The measurements, with symmetric and asymmetric static bandwidth conditions, define the effective advantages and limitations of MP-DPA in terms of transmission cost, which can be adapted by tuning the so-called deadline sensitivity α. The extensive trace-driven studies in highway, rural, and city locations in the North Rhine-Westphalia state in Germany, validate MP-DPA improvements also in a dynamic context. For simulation purposes, are considered two German network providers: O2 (Telefonica) assigned to the preferred interface and Telekom, considered more expensive, attributed to the non-preferred one. MP-DPA is investigated for bulk upload, a car-like transmission, and stream-like upload, typical of video streaming contexts.

The results show that, compared with the off-the-shelf MPTCP, MP-DPA minimizes the non-preferred LTE / Telekom usage without compromising the reliability. The driver position influences the MP-DPA effectiveness: in a rural location, where the network coverage is poor, Telekom usage is reduced by 6% and 4% compared to the baseline MPTCP, with bulk and streaming upload, respectively. The best performances are obtained in urban locations where savings percentages are increased by up to 16%.

The thesis concludes with an overview of the results obtained through realistic trace-driven experiments and an outlook on possible future works on MP-DPA features and implementation modifications.

1 Introduction

In the past years, plenty of companies and organizations estimate that the market of connected cars is going to grow. Following the definition of [2] a connected car "is a vehicle that can communicate bidirectionally with other systems". This allows the car to share internet access, and hence data, with other devices, inside and outside the vehicle (Local Area Network (LAN)). MAVOCO, one of the leading software manufacturers of Connectivity Management Platforms (CMP) [3], states that their number is expected to increase by 270% until 2022 and more than 125 million connected cars will be shipped to customers.

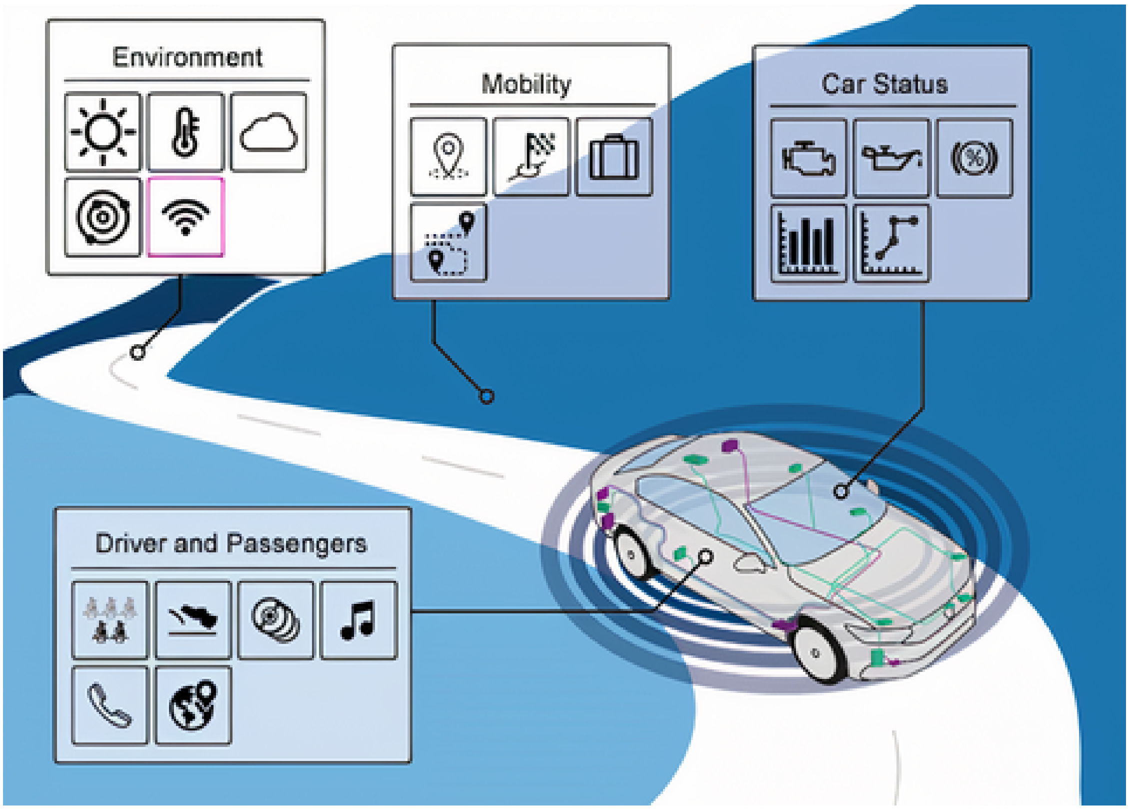


Figure 1.1: Examples of data acquired by sensors. The car continuously senses internal and external information [4].

For the thesis purpose, the scenario of interest is the vehicle-to-cloud, where the technology exchanges information about and for applications of the vehicle with a cloud system. This allows the vehicle to use/provide information from/to others, through the cloud-connected industries like energy, transportation, and smart homes [5].

Another approaching issue that must be faced is that, as the modern vehicles move in their sur-

roundings, they perceive various aspects (environment, mobility, driver and passengers, vehicle status) and collect multiple data via complex on-board sensors as represented in Figure 1.1. With new technologies, the amount of acquired data is strongly increasing, and thereby cars are becoming highly suitable for acting as a mobile sensor network in the IoT context [4].
Consequently, the amount of data produced by every single car grows. As the AutoMat [6] project suggests, instead of using these data only for the correct behavior of the car itself and then discard them, they can be harvested and aggregated to be exploited by external companies for the automotive and non-automotive market.
Since the computing systems of vehicles have low processing power compared to remote servers, the implementation of advanced data processing may be difficult "on-site" (e.g., in the vehicle). Indeed, as a large amount of on-line data may be available (∼4-10 TB/day for autonomous cars [7]), transferring, accessing, and processing this data in real-time with sufficient quality, might be difficult using only the current vehicle computing systems. This arises the challenge of transferring and processing large amounts of data as well as fulfilling the Quality of Service (QoS) requirements like latency and reliability.
The adopted solution consists in leveraging multiple data links with Multipath Transmission Control Protocol (MPTCP) (Figure 1.2) and hereby improving network performance, especially in areas where one mobile network operator lacks coverage. Mostly, in the car-to-cloud scenario, this topic is particularly critical as the vehicles are free to move in the whole territory with completely different network conditions. For instance, the network operators' coverage, as well as the data rates, are not stable and uniform in zones such as highways, smart cities, or rural villages.

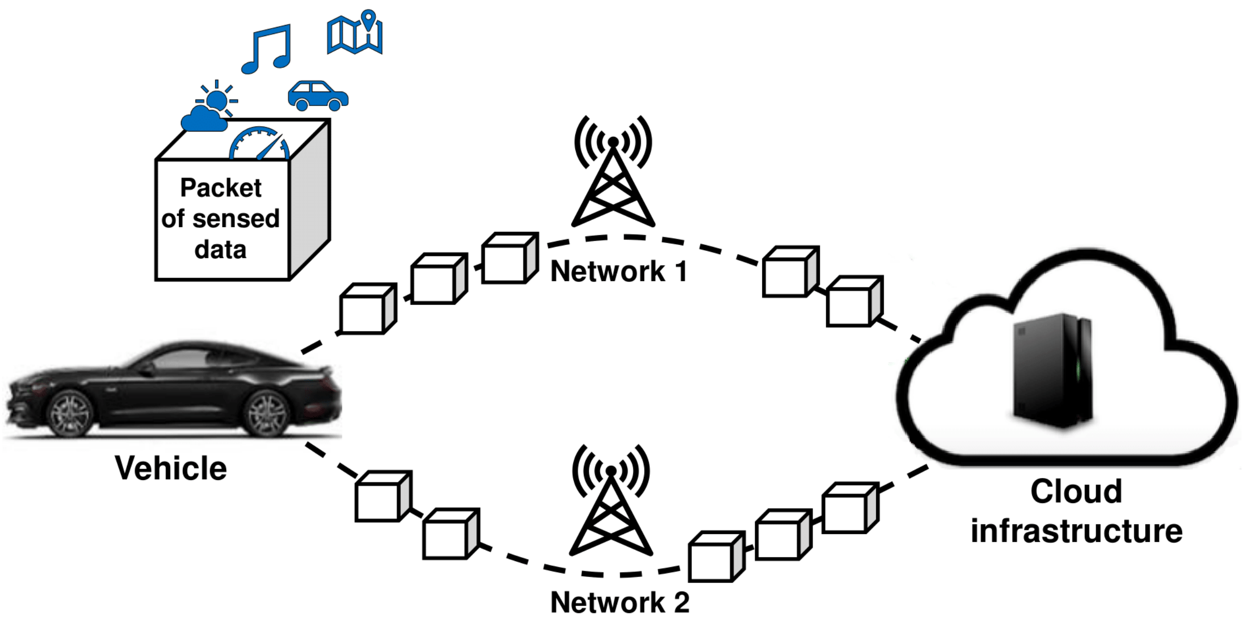


Figure 1.2: Schema of an MPTCP transmission based on packets from a car provided by two interfaces to a cloud. Each packet contains the internal or external data acquired by the car sensors.

The focus of the thesis is on the implementation and evaluation of a new network communication framework based on the multilink protocol MPTCP in a car-to-cloud scenario. The main idea is to replicate MP-DASH presented in [1] and apply it for a dynamic car-to-cloud context for multiple data upload types such as bulk and streaming.

Since the MP-DASH initial goal is to enhance MPTCP to support adaptive video streaming under user-specified interface preferences, it can be perfectly used in a heterogeneous context such as the automotive one. The user-preference aware feature of standard MP-DASH can be exploited in zones where network coverage and data rates are, respectively, not uniform and stable.

Thus, MP-DPA is implemented to be a ”smarter” protocol than MPTCP, able to schedule properly the data in a non-heterogeneous context such as the automotive one. In detail, as its name suggests, Multipath Deadline and Preference Aware (MP-DPA) is implemented as a deadline and preference/cost-aware protocol. The transmission is structured with packets of data which must be uploaded respecting the deadline while minimizing the usage of the non-preferred interface. In this way, if fewer data are sent through the most expensive network provider, with the same reliability performance, the total transmission cost is reduced.

In the first part of the thesis, the research context is defined with a background chapter focused on explaining the fundamentals of already existing transmission protocols such as TCP, MPTCP, and MP-DASH. Then, in the second and more practical chapters, is discussed the basic implementation of the MP-DPA algorithm through the Linux MPTCP kernel module. The three main components that build MP-DPA and their relative procedures and algorithms are detailed. After the presentation of the methodologies used to validate and test MP-DPA, in the last chapter, the simulation results are illustrated and discussed. MP-DPA is deeply analyzed in standard experiments with static bandwidths and bulk data upload to understand its strengths and weaknesses with a comparison with TCP and MPTCP. Their differences in terms of deadlines missed, data distribution over both interfaces, uploading time, and throughput will be largely commented. Besides, a dedicated section will be devoted to the analysis of the deadline sensitivity α to define a possible pareto-optimal value for multiple scenarios.

In the last part of the chapter, MP-DPA is evaluated in dynamic bandwidth conditions using traces taken from real driving measurements. The effectiveness of MP-DPA is tested for highway, rural, and city using multiple deadlines. In addition to the standard bulk upload typical of Vehicle-to-everything (V2X) communications, MP-DPA will be also evaluated and validated for a stream-like transmission.

2 Related Work

Exploiting multilink communication protocols in the automotive context is an ongoing research topic in recent years. The authors in [8], in collaboration with the European AutoMat project, integrate a Common Vehicle Information Model (CVIM) into a vehicle simulator and investigate the network traffic within LTE mobile radio cell. Through a newly generated Car-to-Cloud communication traffic model, the amount of data and data rate in different traffic situations, like normal traffic flow or traffic jams, can be modeled. Although the communication traffic model can be used to simulate the data traffic between cloud and vehicles, the authors do not cover the adaptive aspect. It is treated in [9], where is presented an overview of possible applications and their characteristics. Even if in a completely different context such as Unmanned Aerial Vehicle (UAV) for maritime over-sea missions, Güldenring *et al.* [10] have to face with similar constraints such as the high amount of data, high reliability as well as high data rates. These priorities are satisfied through the usage of MPTCP for each network link. Besides considering issues such as wind, range, and UAV speed, the document reports also some considerations about the results of two different schedulers: default and redundant.

[1] widely discuss MP-DASH, the algorithm to which MP-DPA is based on. It highlights that multipath solution, when applied to video streaming, lacks the support to prioritize one path over another. MP-DASH is extensively studied and evaluated at 33 locations in three U.S. states. Compared with off-the-shelf MPTCP, MP-DASH is very effective: cellular usage reduced by up to 99% and radio energy consumption by up to 85% with negligible degradation of Quality of Experience (QoE).

From a more practical view, MPTCP implementation is described in [11]. Raiciu *et al.* extend TCP to enable it to support multiple paths for current applications on today's Internet. They report the implementation details and performance evaluations in emulated and real-world cases of Multipath TCP in the Linux Kernel, with an in-depth comparison with a regular TCP protocol. Besides, some working optimizations concerning middleboxes, memory usage, and receiver buffer are explained through network emulation.

Moving from practical to engineered view, the authors in [12] try to understand in an emulated network environment how MPTCP performances and throughput are affected in practice by Round Trip Time (RTT) measurements and subflow selection. Their results show that path selection and packet scheduling have a large effect on performance and throughput can be improved by slight modifications to the sender buffer.

A very similar case of study is explored in [13] where the performance of multi-path TCP is investigated in the wild. Chen *et al.* are especially interested in understanding how application-level performance is affected when path characteristics (e.g., RTTs and loss rates) are diverse. Their experiments demonstrate that MPTCP provides robust data transport and reduces the variability in download latency.

Instead of finding the best configurations for reaching the best performances with MPTCP, Hurtig *et al.* in [14] and the authors in [15] investigate the problems in adopting a multipath approach. In the first paper, they prove that low-latency communication with good user experience is hard to achieve with MPTCP when a device has network interfaces with asymmetric capacity and delay (e.g. LTE and Wireless Local Area Network (WLAN)). Moreover, the authors present two novel scheduling techniques: the BLocking ESTimation (BLEST) scheduler and the Shortest Transmission Time First (STTF) scheduler. The paper focuses on comparing them with the MPTCP default scheduler with a final evaluation in real-world environments for interactive applications.

As mentioned before, [15] demonstrates that the current MPTCP suffers from two problems. Firstly, MPTCP can reduce the throughput of other TCP connections without bringing any benefit. The second problem instead is related to the excessive aggressiveness of MPTCP towards TCP users. The authors revisit the problem, caused by an excessive amount of traffic transmitted over congested paths, by implementing the Opportunistic Linked-Increases Algorithm (OLIA), in the Linux kernel. Their measurements, simulations, and theoretical analysis indicate that MPTCP with OLIA is as responsive and non-flappy (it avoids random flipping of traffic between multiple good available paths) as MPTCP and it solves problems P1 and P2.

The attention is then moved to the congestion control algorithm adopted in MPTCP. Specifically, in a high-speed driving scenario in [16], is conducted a detailed measurement study comparing TCP CUBIC with Bottleneck Bandwidth and Round-trip propagation time (BBR) – a new congestion control alternative developed by Google. Briefly, the results show that CUBIC and BBR generally have similar throughput, but BBR has significantly lower self-inflicted delays than CUBIC depending on the amount of data that has to be downloaded.

BBR performances are again investigated in [17] always in an automotive context. Pillmann *et al.* extend the Scalable Network Coding (ScalaNC) framework with the novel congestion control BBR in a Hardware-in-the-Loop field test consisting of an LTE base station and channel emulator. Their final results show a quick adaptation to the frequent data rate changes typical of driving scenarios and an effective transfer of high amounts of data from vehicles to the cloud.

The structure of MPTCP allows a single TCP connection to be split across multiple sublinks. Consequently, it improves the reliability and the networked resourced can be used more efficiently. In [18] is available the design of a multipath congestion control algorithm implemented in Linux and successively evaluated for multihomed servers, data centers, and mobile clients. The paper demonstrates that although some congestion avoidance algorithms can be harmful, the new algorithm improves the performances compared to single-path TCP.

MPTCP is still largely discussed by Ford *et al.* in [19]. In addition to higher throughput, the simultaneous use of multiple paths improves the resilience to network failure. The document proposes a set of extensions to standard TCP to support multipath operations. The protocol has the same characteristics of TCP (i.e., reliability), and it provides the features necessary to establish multiple TCP flows across potentially disconnected paths.

In particular, the focus is on the presentation of changes required to add multipath capability to

TCP. Specifically, the attention is on the modifications done for signaling, setting up and managing the so-called subflows, reassembly of data, and termination of the connection.
However, to implement an MPTCP protocol is needed the knowledge of additional aspects. Thus, other documents are used to complement them. For instance, [20] explains the motivations behind the usage of MPTCP and it defines the multipath architecture. Besides, it contains a discussion of high-level design decisions on which this design is based. The congestion control-related issue is covered in [21]. It presents a congestion control algorithm that ensures safety in managing the multiple paths without compromising the behavior of other network users. Finally, in [22] the authors discuss what impact MPTCP will have on applications, what applications will want to do with MPTCP, and as a consequence of these factors, what API extensions an MPTCP implementation should present.
Concerning the differences between network technologies, Huang *et al.* in [23] compare 4G LTE with 3G/WiFi networks to better understand the performance and power characteristics. The authors, through a publicly deployed tool called 4GTest, studied the network performance of LTE networks and compared them with other types of mobile networks. The results highlight that LTE generally shows higher throughput than 3G and WiFi, which has an average of 13Mbps and 6Mbps, respectively. From the energy point of view, the final evaluations, based on real user traces, show that LTE is 23 times less power efficient compared with WiFi and 3G.
For mobile online social networking services, are conducted different studies. Han *et al.* in [24] confirms that 3G networks are currently overloaded, due to the increasing popularity of various applications for smartphones. The studies suggest that offloading mobile data traffic through dedicated communications techniques can be a possible solution to partially solve this problem since its cost is approximately negligible. The authors proposed a solution to reduce the amount of mobile data traffic and facilitate the transmission of information in the emerging Mobile Social Networks (MoSoNets). As a case study, the authors investigate the target-set selection problem for information delivery. Specifically, the studies are focused on how to select the target set with only k users, such that the mobile data traffic over cellular networks is minimized. The simulation results verify the efficiency of three different algorithms (Greedy, Heuristic, and Random) for both synthetic and real-world mobility traces.
The same mobile device studies are complemented in [25] by conducting the first measurement study of mobile web performance over multipath TCP. Through experiments that analyze real web pages with different settings, Han *et al.* found that the upper-layer web protocols can provoke complex interactions with the transport layer. The results show that, since MPTCP helps to reduce Software Package Data Exchange. (SPDY)'s losses and inefficient bandwidth utilization, a multipath transmission always boosts SPDY while it may hurt Hyper Text Transfer Protocol (HTTP). Furthermore, the document provides in-depth explanations of the root causes, as well as recommendations for improving mobile web performance over MPTCP.

3 Background

The following sections provide the background for some basic notions about vehicular communication and the multipath network protocol MPTCP. Afterward, some TCP parameters and the main already existing algorithms for multilink communication are briefly described to understand the whole range of choices that can be done for the implementation of MP-DPA.

3.1 Vehicular Communication

As introduced previously, the emerging Vehicle-to-everything (V2X) summarizes every communication between a vehicle and a second party. In this context, a vehicle can be anything from cars, to airplanes and trains.
As defined in [26] in most cases, V2X communication can be divided into two sub-classes:

- Vehicle-to-Vehicle (V2V) communication that includes direct communication between two vehicles using any kind of wireless transmission channel
- Vehicle-to-Infrastructure (V2I) communication that provides the foundation for cloud-based services [9]

The latest type of communication is based on the information exchange with the vehicles and cellular communication networks or the so-called Road Side Units (RSU)s – fixed communication stations deployed along the road. In the thesis, V2I communication is modeled by a client-server configuration proposing a new adaptive transmission based on the recent Multipath Transmission Control Protocol (MPTCP).
In addition to V2X sub-classes, in literature, there are two types of V2X communication technology depending on the technology being used:

- WLAN-based
- cellular-based

The original V2X communication uses WLAN technology and works directly between V2V and V2I to form a vehicular network. Hence, it does not require any auxiliary communication infrastructure for vehicles to communicate, which is the key to assure safety in remote or little-developed areas where the network performances are poor. This technology is referred to as Dedicated Short-Range Communications (DSRC), which uses the radio communication provided by 802.11p.
More recent, 3rd Generation Partnership Project (3GPP) [27] published V2X specifications based on LTE as the underlying technology. It is generally referred to as Cellular V2X (C-V2X) to

differentiate itself from the previously cited 802.11p based V2X technology. In addition to the V2X communication, C-V2X also supports the Vehicle-to-Network (V2N) cellular communication.
The communication speed is also increasing with new technologies. Chen in [13] suggests that with the emerging population of smartphones and mobile devices, to cope with the tremendous traffic growth, cellular operators have been upgrading their access technologies from the Third Generation (3G) to the Fourth Generation (4G) networks. As Chen claims in [13], "the specified peak speed for 4G services is 100 Mbps for high mobility communication, and 1 Gbps for low mobility communication".
Experimentally, LTE networks differ from WiFi networks since they provide broader signal coverage and more reliable connectivity under mobility. Although cellular data networks are characterized by high reliability and throughput, they come at the cost of increased delay with larger packet RTTs. In many scenarios, WiFi is no longer faster than 4G LTE, and this provides greater incentive to use an LTE-based transmission also for what concerned the automotive context.
For these reasons, during the whole thesis discussion, LTE technology will be the used one. To achieve this, in both static and dynamic cases, appropriate network parameters are set up through MiniNet modules.

3.2 Multipath Transmission Control Protocol (MPTCP) Principles

Multipath Transmission Control Protocol (MPTCP), as its name suggests, is based on standard TCP connections. Transmission Control Protocol (TCP) is defined as a byte-oriented, reliable, and in-order delivery communication protocol. Its main properties and features are deeply described in several papers [28], [29], [30], [18]. Although TCP characteristics are non further discussed during the theses, its performances will be used as a term of comparison during the evaluation of Multipath Deadline and Preference Aware (MP-DPA). However, TCP can be summarized in five mechanisms as follow:

- Connection setup handshake
- Reliable transmission and acknowledgment of data
- Congestion control
- Flow control
- Connection teardown handshake

To allow cheap and ubiquitous Internet access, smartphone providers are equipping devices with multiple network interfaces to enable handover from slow and expensive mobile data networks to fast and free WLANs, when possible. From an automotive perspective, with the approaching of new technologies and modern vehicles, analogies can be found for the smartphone world.
Although the mobile data networks are becoming faster (e.g. with the introduction of the new 5G standard) and cheaper, it is still possible and convenient to leverage multiple technologies simultaneously to increase throughput and robustness against not stable network bandwidths and not

uniform network operators coverage.
Implementing MPTCP does not consist in simply replicating the TCP connection for each subflow. New options have to be added to the protocol, considering that each packet can take different paths with the related changes on the standard TCP mechanisms (i.e. out-of-order data arrival at the receiver, middleboxes able to modify packets) [11].
As discussed in [1], the Linux Kernel MPTCP is one of the most widely used MPTCP implementations beside Apple's cloud-based assistant system Siri. A Linux implementation is composed of 4 main blocks: the meta socket, the path-manager, the congestion control, and the scheduler. The meta socket is the central abstraction of each MPTCP connection while the path manager decides on the creation and removal of subflows. The MPTCP congestion control is responsible for the congestion window sizing based on multiple algorithms that are available in MPTCP Linux Kernel. The last block and the most relevant for the thesis purpose is the scheduler whose main task is to transparently split data among the subflows.

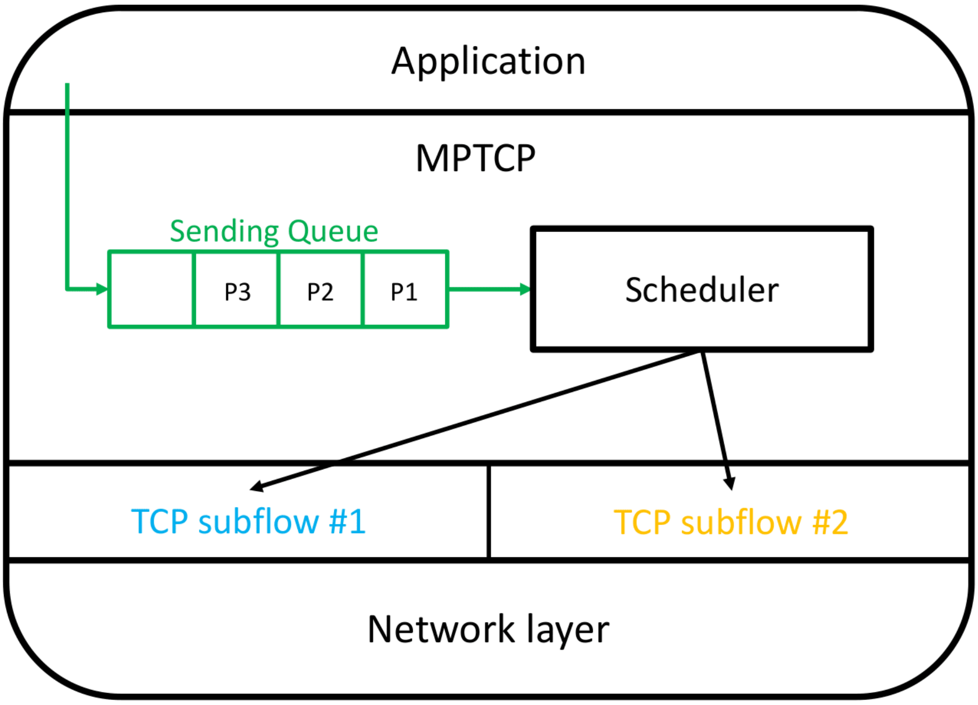


Figure 3.1: MPTCP connections are composed of two TCP subflows. The packets of data that come from the application side are divided into each subflow according to the scheduling policy of the selected scheduler.

As Figure 3.1 schematizes, TCP-based protocols such as MPTCP are implemented as a connection-oriented protocol where initially, a logical connection is created between two endpoints (local and remote) and then, data is transferred in a bidirectional fashion. A logical MPTCP connection is split into subflows (i.e. LTE and WLAN interface), where each one behaves like a traditional TCP connection.
Each data has to be ordered following a sequence number to guarantee an in-order delivery at the receiver. The reliability of the protocol is provided by acknowledgments (ACK) that are sent by

the receiver to the sender. If the sender does not receive ACK after sending data, it has to re-send it again.
The volume of data flowing within a link and to ensure that a sender is not overwhelming a receiver (i.e. control flow) are controlled, respectively, by the congestion window and the receive window. A sender transmits as much data as a receiver can afford through the reading of the receive window. For what concerns the congestion control, TCP uses several mechanisms to achieve high performance and avoid congestion collapse. Performances highly depend on TCP Congestion Avoidance Algorithm (CAA) used.

3.2.1 Connection and Subflow Establishment

For what concerns MPTCP, the first connection establishment is performed in the same way of TCP through the three-way handshake (ACK - SYN ACK - ACK packets). Once an MPTCP connection is established, each end-host can recognize the packets that come from the same peer despite they have different source IP addresses.
As many of the mobile clients are behind Network Address Translationss (NATs), when the server has an additional interface, it is difficult for the server to directly communicate with the mobile client as the NATs usually filter out unidentified packets. For this reason, when the client has an additional interface, it will first notify the server its additional Internet Protocol (IP) address with an *Add Address* option over the established subflow, and then it sends another SYN packet with a JOIN option to the server's IP address. With this MPTCP - JOIN option, this subflow will be associated with a previously established MPTCP connection [13].

3.2.2 Congestion Avoidance Algorithms

One key element of MPTCP is the availability of controlling the congestion window through a list of possible algorithms.
As each MPTCP subflow behaves as a TCP flow, after the 3-way handshake, each subflow maintains its congestion window and retransmission scheme during data transfer. It begins with a slow-start phase that doubles the window per RTT before entering the congestion avoidance phase [13].
The choice concerning which congestion avoidance algorithm can be used for the thesis purpose was done between CUBIC, BBR, and OLIA.

CUBIC [31], which is the most widely used algorithm in Linux operating system, relies on packet losses as an indicator of congestion. The protocol modifies the linear window growth of the standard TCP to be a cubic function to improve the flexibility of TCP. By making the window growth to be independent of RTT, it also achieves uniform bandwidth allocations among flows with different delays.
During steady-state, CUBIC increases considerably the window size when the window is far from the maximum value, and then slowly when it is close. This property allows CUBIC to be very flexible and stable when the bandwidth and latency of the network are not negligible.

Bottleneck Bandwidth and Round-trip propagation time (BBR) [32] was proposed in 2016 by Google as a high-performance TCP congestion control algorithm able to provide high performances especially on a network with high latency and high packet loss ratio. BBR periodically estimates the available bandwidth and RTT to recognize the rates of the current transmission. In other words, its congestion window is derived from an estimate of the communication link's bottleneck bandwidth. The estimate is derived from the amount of acknowledged data as well as the round-trip time.

From the empirical results given by experiments [33], depending on the combination of values for RTT, bandwidths, and buffer size, BBR achieves different performances. Thus, is possible to identify network conditions under which BBR shows higher throughput than TCP congestion control algorithms such as CUBIC. The results of a highway driving condition [16], instead, show that CUBIC and BBR generally have similar throughput, but BBR has significantly lower self-inflicted delays than CUBIC. The same considerations are reported by the authors in [17] who confirm, through TCP simulations in highways, that BBR performs better when packet loss occurs on the communication link with low Signal to Interference and Noise Ratio (SINR).

Opportunistic Linked-Increases Algorithm (OLIA) is a congestion avoidance algorithm designed by Khalili and largely described in [15]. Its main goal consists in solving the not pareto-optimal MPTCP behavior linked to its aggressiveness toward TCP users. Being a pareto-optimal algorithm, OLIA is a responsive (it uses capacity quickly when it is available) and non-flappy algorithm (it avoids random flipping of traffic between multiple good available paths). In other words, OLIA is an MPTCP-oriented congestion algorithm that tries to spread the traffic in an optimal way between different paths.

All three algorithms were simulated and their results concluded that BBR is the most suitable for MP-DPA especially for its reactivity in managing multiple interfaces.

3.2.3 Existing Schedulers

When an MPTCP connection contains two or more subflows, the scheduler decides, for each segment, which one to use for transmission. The decision is typically done considering network properties of the subflows' underlying network paths but it can change following different trade-off. Some scheduling algorithms available in the standard version of MPTCP Linux Kernel such as Default, Round Robin, and Redundant are reported and briefly described.

Below, are revised notable MPTCP schedulers from the related work to highlight the key functionalities and explained which one is considered in MP-DPA.

Round Robin Scheduler

Round Robin scheduler [34], [35] is a simple scheduling mechanism. There is no priority between subflows but it selects them in a round-robin fashion one after the other and skips the ones with an exhausted congestion window. This approach should guarantee that the capacity of each path is fully exploited and the distribution of the data packets among all the subflows is equal. Its performance is not optimal and it is only interesting for academic or testing purposes.

(Default) Minimum RTT Scheduler

The Minimum RTT algorithm is the current default scheduler in the MPTCP Linux Kernel [11], [35]. It schedules the last packet of the sending queue to the subflow with the lowest Round Trip Time (RTT) and non-exhausted congestion window. Then, the next packet of data is sent on the subflow with the next higher RTT. Minimum RTT scheduler is particularly effective in heterogeneous networks where scheduling data to the subflow based on the lowest RTT is beneficial since it improves the user experience.
In the same way as the round-robin scheduler, as soon as all congestion windows are filled, the transmission over that subflow is blocked. When a packet is correctly received, the receiver sends the acknowledgment to make free the congestion window and thus allow the scheduler to transmit again data over the same interface. As a negative aspect, the Minimum RTT scheduler makes the sub-flow with good better performances hold more data.

Redundant Scheduler

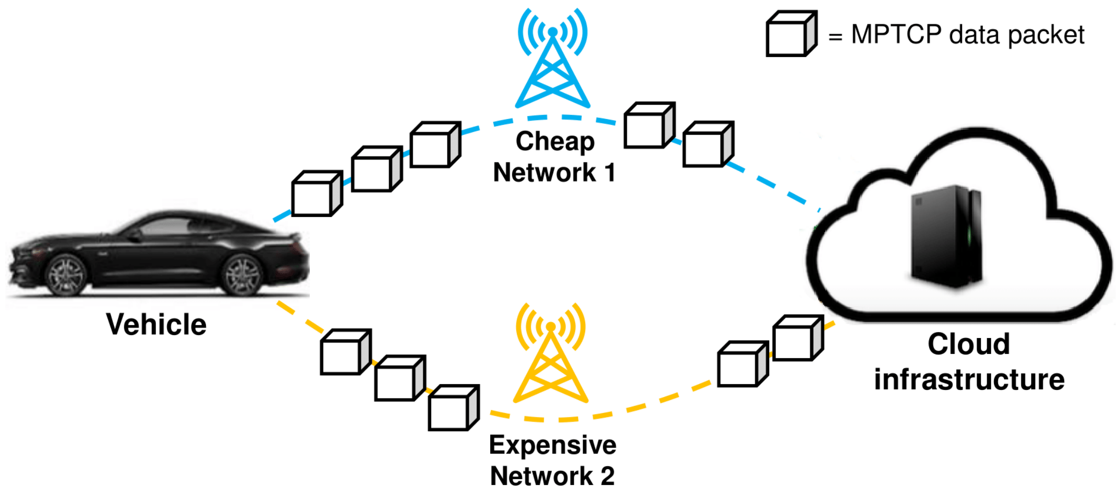
The redundant scheduler [36], [37] was born to face with heterogeneous networks that operate under uncertainties as traffic load variation and failures. Redundant packet scheduling techniques try to face these uncertainties to sustain the best transmission in terms of throughput and latency for delay-sensitive applications, such as online gaming and streaming. The main principle consists in replicating data packets on multiple paths to mitigate the negative effects of heterogeneity such as fluctuations in delay, loss rate, and bandwidth. From the opposite side, it can reduce the advantages of an MPTCP transmission like the overall throughput.
From the thesis prospective, the default Minimum RTT scheduler will be exploited given its flexibility and its notoriety.

4 Design and Implementation of Multipath Deadline and Preference Aware (MP-DPA) Vehicle to Cloud Communication

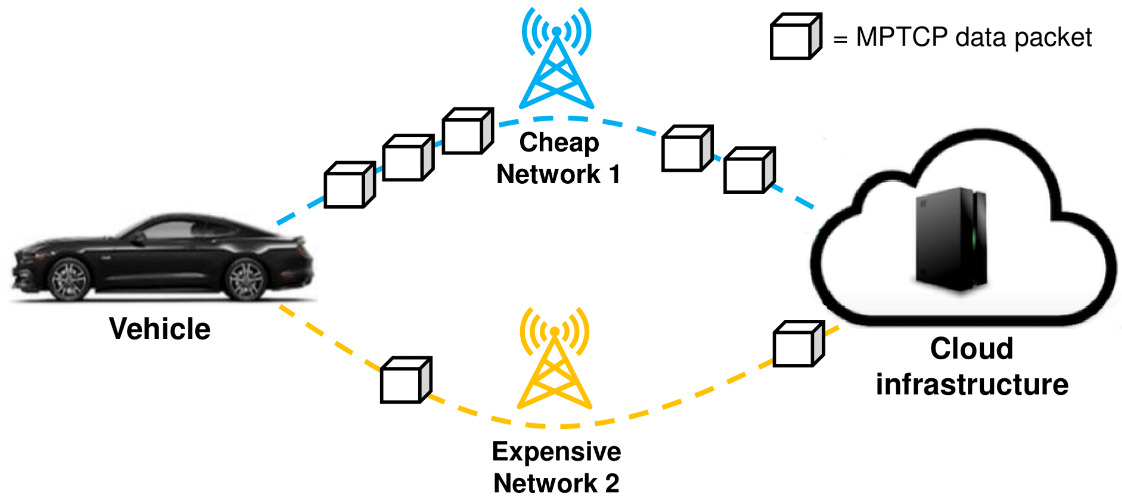
The new Multipath Deadline and Preference Aware (MP-DPA) is implemented and tested on an MPTCP Linux kernel v5.4.0 machine. In the following and more practical sections, after defining MP-DPA concepts and the main TCP parameters used during simulation, is presented the coding structure of the major components: sender and receiver.

4.1 Definition of MP-DPA

As explained before, MPTCP is a standardized multipath solution allowing applications to transparently use multiple paths that are utilized following decisions of the scheduling algorithm.
Han *et al.* in [1], states that the de-facto multipath solution lacks the support to prioritize one path over another. The whole paper is focused on video streaming and, when MPTCP is applied to them, it may cause undesired network usage such as substantial over-utilization of one subflow over the other.
They propose a novel multipath video streaming framework Multi Path Dynamic Adaptive Streaming over HTTP (MP-DASH), a multipath framework for video streaming with the awareness of network interface preferences from users. The basic idea behind MP-DASH is to enhance MPTCP by scheduling strategically video chunks to support adaptive video streaming under user-specified interface preferences.
The solution proposed in this thesis and based on MP-DASH is called Multipath Deadline and Preference Aware (MP-DPA). It is designed to be a framework that uses the multipath concept to schedule properly the data in a non-heterogeneous context such as the automotive one, where the current network bandwidth and network provider coverage change depending on the vehicle position and environment. In detail, instead of tuning the scheduler with respect the user preferences like MP-DASH, MP-DPA, through a goodput estimation process, has to automatically recognize if it is worth to enable or disable the transmission over the second (and most expensive) interface (Figure 4.1).
In this way, since the usage of the most expensive interface or network provider is minimized, globally, the transmission is less expensive than standard multipath solutions while maintaining similar performances such as reliability.
Of course, the new MP-DPA will not be used anymore for streaming video but it will be applied for data transmission in the automotive context for burst and streaming upload following the chunks-



(a) MPTCP does not prioritize one path over the other during the packet transmission. In case the networks have same performances, the packets are divided equally among the subflows.



(b) MP-DPA schedules each packet depending on the network characteristics. If the network conditions are favorable, the usage of the most expensive subflow is reduced.

Figure 4.1: Representation of a multilink car-to-cloud communication based on packets using Multipath Transmission Control Protocol (MPTCP) **(a)** and Multipath Deadline and Preference Aware (MP-DPA) **(b)**.

delivery schema such as MP-DASH.
The overall cost of the transmission and the respect of deadlines in uploading packets of data will be the metrics of main interest. It is easy to understand that these metrics are inversely proportional. A lower amount of data sent over the second interface reduces the transmission cost but can compromise the reliability of the scheduler causing a high percentage of non-respected deadlines. In the opposite case instead, by enabling more often the non-preferred interface, more likely the deadlines are respected at the expense of a higher cost.
It should be noted that in case the preferred interface can sustain alone a reliable transmission of every packet, the non-preferred interface will be permanently disabled and conceptually, MP-DPA

results in a simple TCP connection.

4.2 TCP Parameters Configuration

Moreover, additional secondary parameters can be configured during simulation depending on the scheduler and CAA. Several kinds of research have been done regarding the behavior of these TCP variations as mentioned in 3.2 and, for this reason, the thesis does not expand these topics.
Since the Minimum RTT scheduler is the default TCP scheduler implemented in the Linux Kernel, all the performed experiments refer to it. An analog choice was done also for deciding which CAA will be used in all the cases of study. After some tests focused on understanding which CAA supports in the best way the new MP-DPA, the obtained results show that the Bottleneck Bandwidth and Round-trip propagation time (BBR) algorithm 3.2.2 is slightly better than to CUBIC and OLIA, given its cleverness and its estimation mechanism which makes the transmission less bursty.
In parallel to the CAA choice, the TCP buffer of both sender and receiver is manually set to a fixed value of 2 MB via sockets Application Programming Interface (API) (SO_SNDBUF, SO_RCVBUF).
As largely discussed in [12] [38], the buffer size choice has to take into account the context in which TCP is exploited. In the case of MPTCP, very small send buffers may prevent it from efficiently scheduling data over different subflows, thus, it does not make sense to use a multiple path protocol in combination with small send or receive buffers.
As done with other main parameters, also for the sender and receiver buffers are conducted several short experiments to investigate which size can sustain the maximum bandwidth in every experiment. Thus, since in the thesis context the maximum achievable speeds during static experiments do not exceed 16 Mbps, a sender buffer size of 2 MB is considered as sufficient. However, during dynamic experiments, the bandwidth can reach also higher peaks of 50 Mbps, and thus, the buffer must be increased accordingly to 64 MB.

4.3 Implementation Details and Code Structure

The whole project consists of ∼700 lines of Python code. Its purpose is to simulate a classic exchange of data between client and server with different network configurations, data upload techniques, and communication protocols.
Each experiment is set up through the MiniNet module [39] and standard Linux commands on Internet Protocol Version 4 (IPV4), while the data are transferred from a sender (vehicle) to a receiver (cloud) using the socket module and its APIs [40].

4.3.1 Sender and Receiver

Both sender and receiver scripts can be executed in an arbitrary way depending on the network configurations, communication protocol (TCP, MPTCP, or MP-DPA), and type of experiment (bulk or stream) that has to be simulated.
Thus, some standard network parameters, such as sender and receiver IP addresses, socket port, and buffers size, can be modified to adapt each simulation to the user preferences. Being the sender more

complicated than the receiver, it is provided by additional options like the used protocol, simulation time, data upload type, packet dimension, and time interval for the MP-DPA estimation process. For completeness, the parameters that can be manually configured at simulation time are briefly commented below.

- *receiver IP* and *PORT* to connect to the receiver socket
- boolean values *MPTCP* and *MP-DPA* to define the protocol used
- *time* to define the simulation time
- *buffer size* to configure the sender/receiver buffer size
- *interval* to define the time interval used for the throughput estimation
- *stream upload* to enable the streaming upload

From the coding point of view, as sender and receiver are implemented as a client-server application based on sockets [41], the latest acts as server and waits for connections from the client/sender following the schema reported in Figure 4.2.
The left-hand column represents the receiver, while on the right-hand side is the sender. Starting from the top left-hand column, the server makes to set up a listening socket (*socket(), bind(), listen(), accept()*).
A listening socket with its related IP address and the PORT number does just what it sounds like: it listens for connections from clients.
When the sender calls *connect()* to establish a connection to the server, the three-way handshake is initiated. In this step, the buffers for server and receiver are manually set to a big enough value equal to 2 MB to ensure the high-performance transmission typical of multilink protocols (4.2).
In the middle is present the transmission loop, where data are exchanged between the client and server using calls to *send()* and *recv()* to send and receive packets of data, respectively. At the bottom, after the full transmission of a packet, the client and server close their respective sockets. In case a packet is not fully transmitted due to the expiration of its deadline, the transmission is interrupted and the next packet is sent from scratch.

4.3.2 Sending Procedure

Till now, the document was mainly focused on the standard TCP and network parameters from a high-level point of view. But how is the sender structured is such a way that the new MP-DPA scheduler can establish a reliable and low-latency connection based on a sort of estimation? And how can MP-DPA minimize the use of the non-preferred interface without compromising its constraints?
MP-DPA is designed as a deadline-aware and reliable scheduler that takes into consideration the interface preference, quantified by the cost associated with each interface.
In the thesis context, the reliability of the scheduler is expressed by the number of packets that are entirely sent before their deadlines expire. The reliability aspect is taken into account by the

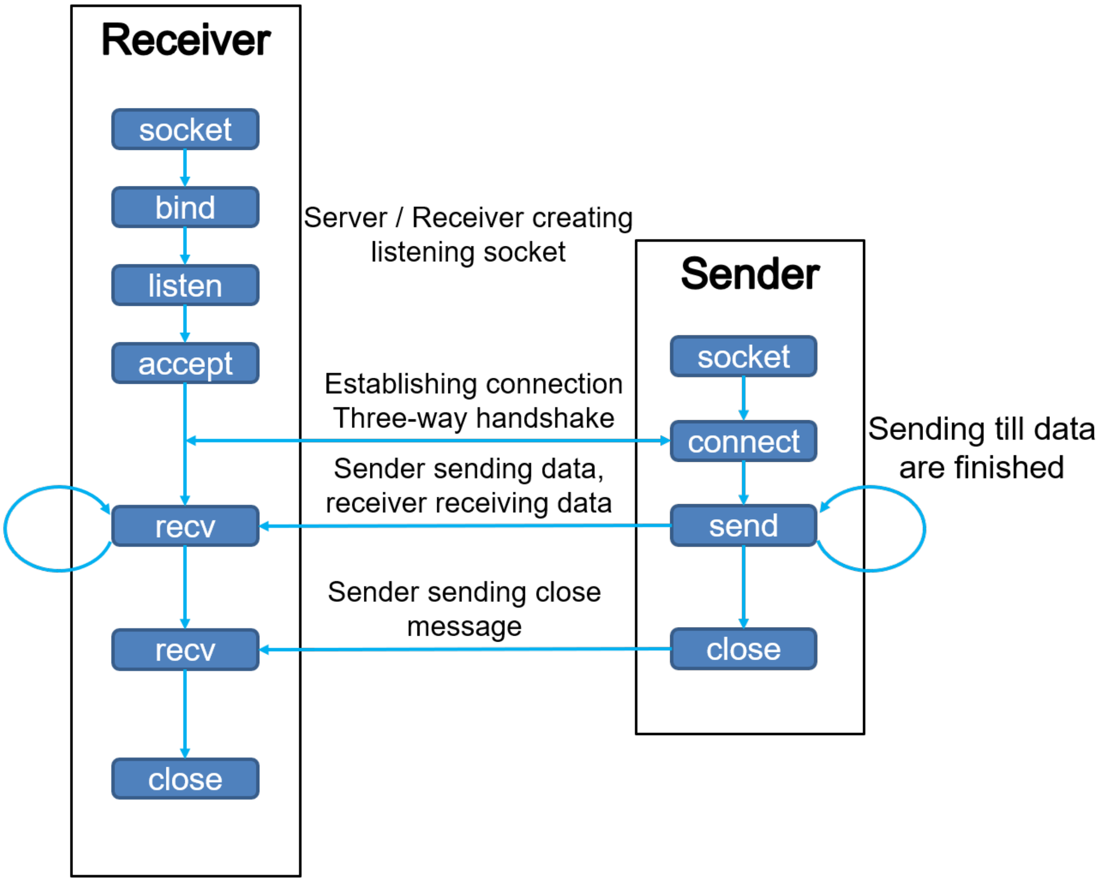


Figure 4.2: Standard TCP socket flow for a client-server communication. After the receiver started listening for possible transmissions from clients, the sender establishes the connection. When the client finished the packet transmission, it sends a close message to the server.

sender that estimates the effective goodput at the application layer in a given time. The result is useful to understand if in the current conditions (single or double interfaces enabled) the whole packet can be sent respecting the deadline. Through this check, the non-preferred interface can be enabled or disabled.

This behavior can be efficiently exploited to limit the overall cost of a transmission. For instance, in the context of a vehicle provided of multiple network interfaces, generally are desired cheap transmissions of data packets. In case the second interface is more expensive than the first one (e.g. Telekom), standard multipath protocols like MPTCP transmit the data on both interfaces using the full bandwidth. Very high performances can be achieved at the expense of an elevated transmission cost.

In case MP-DPA is exploited instead, the goal is to minimize the usage of the most expensive interface while having similar performances and maintaining the transmission reliable.

General Formulation

Here is provided a general formulation of the MP-DPA scheduling algorithm.
Let N ≥ 1 be the number of network paths. Let S be the chunk size and D be its upload deadline. The duration of each time slot is d. Let $b(i,j)$ be the available bandwidth of interface i at time slot j ($1 \leq i \leq N, 0 \leq j < D$), and $c(i,j)$ be the unit-data cost of using interface i at time j.
x(i,j) is a binary decision variable where $x(i,j) = 1$ iff the interface i is used at time j, otherwise $x(i,j) = 0$. The goal is to minimize the overall cost

$$C = \sum_{1 \leq i \leq N, 0 \leq j < D} c(i,j)b(i,j)x(i,j)d$$

under the constraint that the deadline is met:

$$\sum_{1 \leq i \leq N, 0 \leq j < D} b(i,j)x(i,j)d \geq S$$

The theoretical approach consists in choosing items such that the deadline is met and the total cost is minimized. The algorithm tries to solve the problem using an heuristic procedure based on estimation.

Practical Online Procedure

In practice, MP-DPA needs to run in an online fashion where the goodput estimation is continuously updated. In every experiment, the hosts have 2 interfaces (N=2) and the first interface is always preferred over the second one. Thus, $c(preferred_LTE, j) < c(non_preferred_LTE, j)$ for ∀j.
MP-DPA leverages the existing MPTCP schedulers to distribute packets over multiple paths and adds the intelligence of how to control the non-preferred subflow. Essentially, MP-DPA introduces both deadline-awareness and priority into these schedulers.
The online version of MP-DPA sending procedure is depicted in Algorithm 1.
It has three input parameters, the video chunk size S, the length of upload time window D (the deadline), and the time after that a new estimation is needed. At the beginning of the connection, MP-DPA drives the preferred subflow to its full capacity and turns off the non-preferred one. It then monitors the progress of data transfer and, if necessary, turns the non-preferred interface on. It happens when the preferred interface underperforms and the deadline would be missed if the scheduler continues using only the first LTE.
The while loop is responsible for sending the chunk's data using MPTCP (line 11). Est_thr at line 17 is the current estimation of the actual goodput that is computed after every $timeIntervall$ seconds. Its value is exploited by MP-DPA to checks if the preferred LTE alone is sufficient to transmit the remaining data of the packet and disables the seconds subflow (lines 19–21). Since the throughput may change over time, it also needs to check if the non-preferred subflow should be enabled again to sustain the upload (lines 22–24).
To be conservative and compensate for the estimation inaccuracy of the throughput, α, the deadline sensitivity can be configured to be less than 1 in lines 19 and 22. The smaller the value of α is,

Algorithm 1 Sending procedure

```
input: S − video chunk size, D − deadline window, timeIntervall = 1ms;
sentBytes = 0;
timeStart = gettimeofday()
while (sentBytes < S) do
    if S − sentBytes > packetSize then
        n = packetSize;
    else
        n = S − sentBytes;
    end if

    Get n bytes from the remaining data and send them using MPTCP;
    sentBytes+ = n;
    timeNowt = gettimeofday()
    timeSpent = timeNow − timeStart;

    if timeSpent > timeIntervall then
        Estimate the current throughput est_thr;

        if (α x D - timeSpent) x est_thr > (S - sentBytes) then
            Disable non-preferred interface;
        end if
        if (α x D - timeSpent) x est_thr < (S - sentBytes) then
            Enable non-preferred interface;
        end if
    end if
end while
```

the less likely the actual deadline is missed. However, a smaller α also leads to more data over the non-preferred interface. The function of α will be largely described with some experiments aimed at finding possible pareto-optimal values of it for different network conditions.

5 Methodologies for Evaluation Framework and Scenario Description

All the experiments are set up on an HP Pavilion Notebook with an Intel Core i5 processor, 2.30 MHz frequency. Everything is implemented in Ubuntu 18.04.4 with a 64-bits MPTCP Linux kernel v5.4.0 virtualized by Virtual Box. Both sender and receiver are modeled in Python 3 and executed through Visual Studio Code v1.43 editor. For what concerned the evaluation and plotting sections, Jupyter Notebook is used for a graphical representation of the results. The network environment is emulated through the open-source MiniNet emulator v2.2.1 while the throughput in every experiment is monitored with Bandwidth Monitor NG (bwm-ng) v0.6.2.
This chapter introduces the methodology for the system evaluation and the underlying evaluation framework. In particular, in the next sections, are introduced the metrics of interest for MP-DPA evaluation and the setup used during each experiment. The chapter concludes with some details on the simulation schema adopted for static and dynamic scenarios.

5.1 Performance Metrics

The main goal of the experiments is to analyze if the new MP-DPA can optimize the standard multipath protocols in multiple bandwidths and data upload conditions. It aims to reduce the usage of the non-preferred interface to reduce the overall transmission cost, without compromising the MPTCP performances such as high throughput and reliability. For this purpose, the concept of deadline is introduced.

5.1.1 Reliability and Interfaces Usage

The deadline is defined as the time within which a package must be sent in its entirety. MP-DPA reliability is expressed as the number of packets that are not completely sent before their deadline expires. Thus, for every experiment will be presented the percentage of missed deadlines for TCP and MPTCP results.
Besides satisfying every deadline, MP-DPA tries to minimize the exploitation of the non-preferred interface. It is achieved by dynamically enabling and disabling the second sublink based on the results of the throughput estimations at application layer. Also is this case, the percentage of data sent over both interfaces will be analyzed in parallel to TCP and MPTCP.
It is easy to understand that the previous metrics are in contrast with each other. By achieving a very low percentage of deadlines missed, the data sent over the non-preferred interface is more likely to be high. In the opposite case instead, a lower amount of data sent over the second interface can

compromise the reliability of the scheduler causing a high percentage of non-respected deadlines. In case the preferred interface can sustain alone a reliable transmission of every packet, the previously described scenarios will not happen. In that scenario, the second interface will be permanently disabled and conceptually, MP-DPA results like a simple TCP connection.

For this reason, there is a need for a parameter that interconnects these metrics and compensates for throughput estimation inaccuracy. As introduced in 4.3, the so-called deadline sensitivity α can influence the MP-DPA performances in terms of missed deadlines and data sent over the non-preferred interface and thus overall throughput. It directly operates during the throughput estimation by changing the deadline seen by the scheduler (virtual deadline). With $\alpha = 1$, the virtual deadline and the one set by the user (real deadline) coincide. In this case, the scheduler works as if the user has no control over it. α can be also decreased below 1. A smaller value of the deadline sensitivity reduces the virtual deadline and the scheduler will enable the non-preferred interface more often. As a consequence, less likely the real deadline will be missed. With $\alpha > 1$ the scheduler operates oppositely. Therefore, the deadline sensitivity does not provide one value which suits best for each scenario. To identify the best overall value, one sensitivity analysis is conducted in section 6.5.

5.1.2 Latency and Throughput

Being a communication protocol, MP-DPA is also analyzed from the throughput and latency point of view. The real throughput is measured for both interfaces during the whole simulation time, investigating the differences between MPTCP and TCP. Furthermore, will be considered the average time needed to entirely send a packet with particular emphasis on the adaptive behavior of MP-DPA. Exploiting MP-DPA, the time for a packet decreases linearly with a lower value of α and a stricter deadline. MPTCP instead experiences a fixed transmission time since it always uses the maximum speed.

5.1.3 Utility Factor

In addition to the standard metrics, MP-DPA is in-depth analyzed to find the best deadline sensitivity α that fits for a given network configuration. It takes the name of deadline sensitivity analysis and will be defined by the utility factor μ which expresses how much MP-DPA is effective with a given α.

Changing the priority of the metrics that compose μ is possible to define different utility factors and thus analyze MP-DPA under different perspectives. For this purpose, in the thesis are considered two utility factors: one is more focused on the reliability aspect of MP-DPA while the second one is more centered on the non-preferred interface usage minimization.

The generic formula used for the utility factor computation is based on the combinations of the main MP-DPA features.

Given three weights w_1,w_2 and w_3,

$$\mu = w_1 * d + w_2 * l + w_3 * o$$

where:

- d is the percentage of deadline missed (6.5.1)
- l is the percentage of data sent over non-preferred LTE (6.5.1)
- o is the percentage of MP-DPA overhead with respect the total amount of data sent with MPTCP (6.5.2)

such that

$$0 \leq \mu \leq 1 \qquad \forall d, l, o$$

The weights reported in Table 5.1 are set in such a way that μ assumes a value between 0 and 1. In the best case, when μ=1, no deadlines are missed, no data are sent over the non-preferred interface and no overhead is present with respect to MPTCP.

	Target	$\mathbf{w_1}$	$\mathbf{w_2}$	$\mathbf{w_3}$
$\boldsymbol{\mu}_1$	Reliability	0.60	0.25	0.15
$\boldsymbol{\mu}_2$	Non-preferred interface usage minimization	0.10	0.80	0.10

Table 5.1: Weights for utility factors computation.

5.2 Setup of Experimental Validation using MiniNet Network Emulator

The adopted network setup is designed to allow the simulation of experiments in a controlled MiniNet environment with different characteristics. Figure 5.1 illustrates the emulated topology realized with MiniNet.

The model consists of a sender (vehicle) and a receiver (cloud) connected through a simple network composed of multiple links and switches.

For measurements with standard TCP, is set up the 2-path scenarios (solid lines). In this case, the client and the server have just one active Ethernet interface, assigned to an IP address 10.1.1.10 and 10.1.1.20, respectively. The link between Switch 1 and Host 2 has a fixed high bandwidth of 1 Gbps and approximately zero latency/delay, so as not effect bottleneck bandwidth nor total latency.

The interesting link is the one that connects Host 1 and Switch 1. Its bandwidth can be dynamically configured with a fixed value for the whole simulation (static experiments) or changed at run time (dynamic experiments). Since the carrier's commercial LTE network have a latency of ∼50 ms [1], the delay during simulation is set to the same value. In this way, the total Round Trip Time (RTT) on the whole path is approximately equal to 100 ms.

In the case of experiments that include the usage of multipath protocols, the previous path is mirror duplicated (dashed-line in Figure 5.1). Hence, a new Switch is added to the topology such as new links that connect Host 1 and Host 2. Both hosts must be equipped with a second interface, with

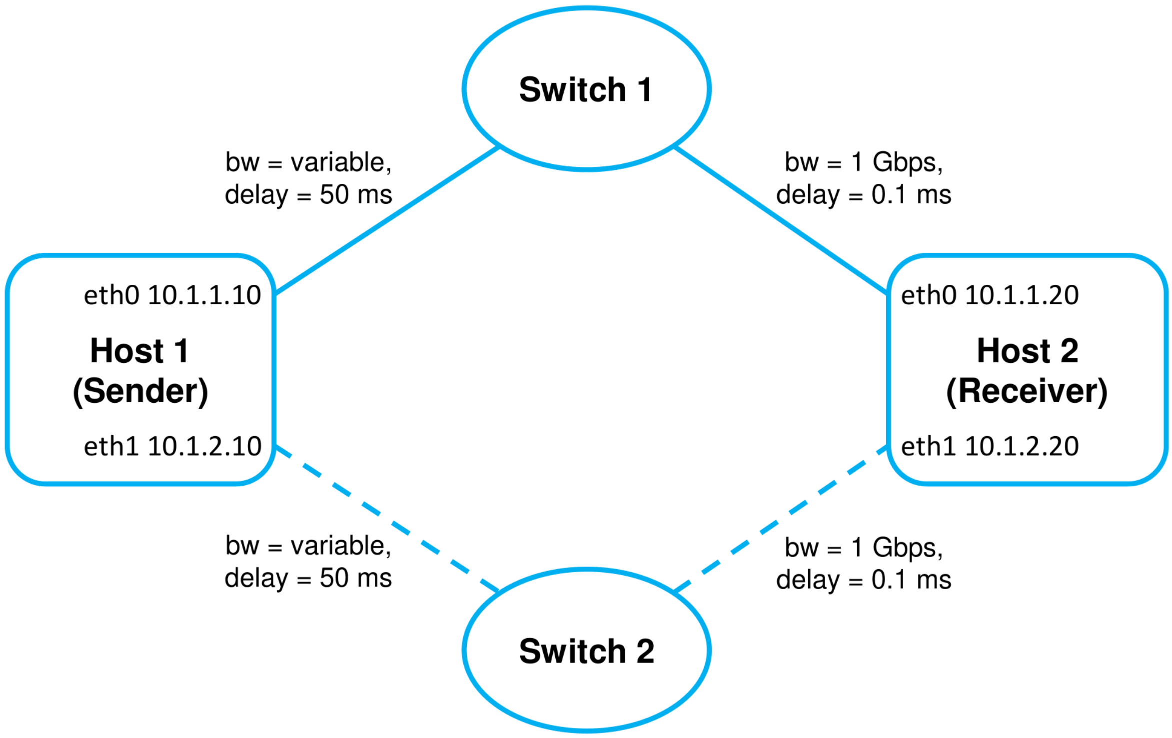


Figure 5.1: Network topology emulated by MiniNet. For TCP experiments both hosts are provided by only one interface (solid-line paths). The additional dashed-line paths are included for the multilink experiments (MPTCP and MP-DPA). Both links at the sender side are characterized by configurable bandwidth and fixed delay.

the same properties of the first one and with IP addresses 10.1.2.10 and 10.1.2.20 for the sender and receiver, respectively.

To configure both interfaces IPs with specific addresses and enable a multipath communication, is needed a manual configuration. In fact, with multiple addresses defined on several interfaces, MiniNet must be able to tell the kernel that if a source address is selected, to use that specific interface + gateway and not the default ones. Thus, is arranged one routing table per outgoing interface where each routing table is identified by a number. The configuration procedure of the routing tables is available at [42].

For all the experiments, the links are set to have no losses and the jitter is not considered. However, they can be arbitrarily changed for future works.

As the network topology and its basic parameters can be set up at will, can be initiated in the emulated environment different types of connections between sender and receiver:

- single-path TCP: sender and receiver activate only their primary Ethernet interface (eth0).
- double-path MPTCP: sender and receiver activate both Ethernet interfaces (eth0 and eth1).
- double-path MP-DPA: same configuration of MPTCP but with some additional features on the sender side.

The differences between these protocols, in terms of amount of data sent over the non-preferred interface and percentage of missed deadlines, will be highlighted in different network conditions.

5.3 Network Emulation Environment

For network emulation environment is intended the wrapper script that creates the network topology (5.2) provided by hosts, switches, and links. It has also to invoke receiver and sender scripts and monitor the effective throughput for the evaluation phase.
The same dynamic structure present in sender and receiver scripts is also defined for the emulation environment to guarantee an arbitrary execution of experiments. At execution time, multiple parameters can be manually set:

- boolean values *MPTCP* and *MP-DPA* to define the protocol used
- network secondary parameters such as *loss* and *jitter*
- *stream upload* to enable the stream data upload
- *time* to define the simulation time
- optional *debug* to enter in debugging mode through WireShark [43]

Despite the emulation environment can be simulated in numerous ways with different combinations of parameters, for this work, the focus is shifted on ideal networks with neglected jitter and data loss percentages. For the whole experiments, the flow followed by the wrapper script is approximately the same with some small changes between static and dynamic scenarios. Figure 5.2 schematizes this structure highlighting the main operations and differences between the 2 study cases. The left stream represents the process in a static environment with fixed bandwidth while the right side concerns the dynamic one.
Starting from the left, the first step consists in designing the desired network topology. Depending on the choice of simulation, it can be composed of multiple switches and hosts as Figure 5.1 depicts. Here, every link is configured properly with its fixed bandwidth.
Can be seen that till now, are not provided information about the bandwidths of each interface. In addition to the deadline value and the deadline sensibility α, their values are provided by a modifiable CSV format file. The motivation behind this choice simply lies in the flexibility to simulate the network and in an easier way to evaluate the results. Depending on the experiment, the CSV format file contains the value of the deadline for stream and bulk data upload, the α value, and the interfaces bandwidths for static conditions simulations.
For what concerns ipv4 standard parameters configurations, are exploited shell commands like *sysctl*. In particular, for all the experiments BBR (3.2.2) is set as Congestion Avoidance Algorithm (CAA) while the Minimum RTT algorithm as scheduler (3.2.3).
In case MPTCP is enabled and thus multiple addresses are defined on several interfaces, the IP configuration step defines one routing table per outgoing interface. At this point that the network environment is defined and configured, the sender and receiver can be invoked to emulate a transmission of data. The followed process for both stream and bulk upload is the same. Firstly, the

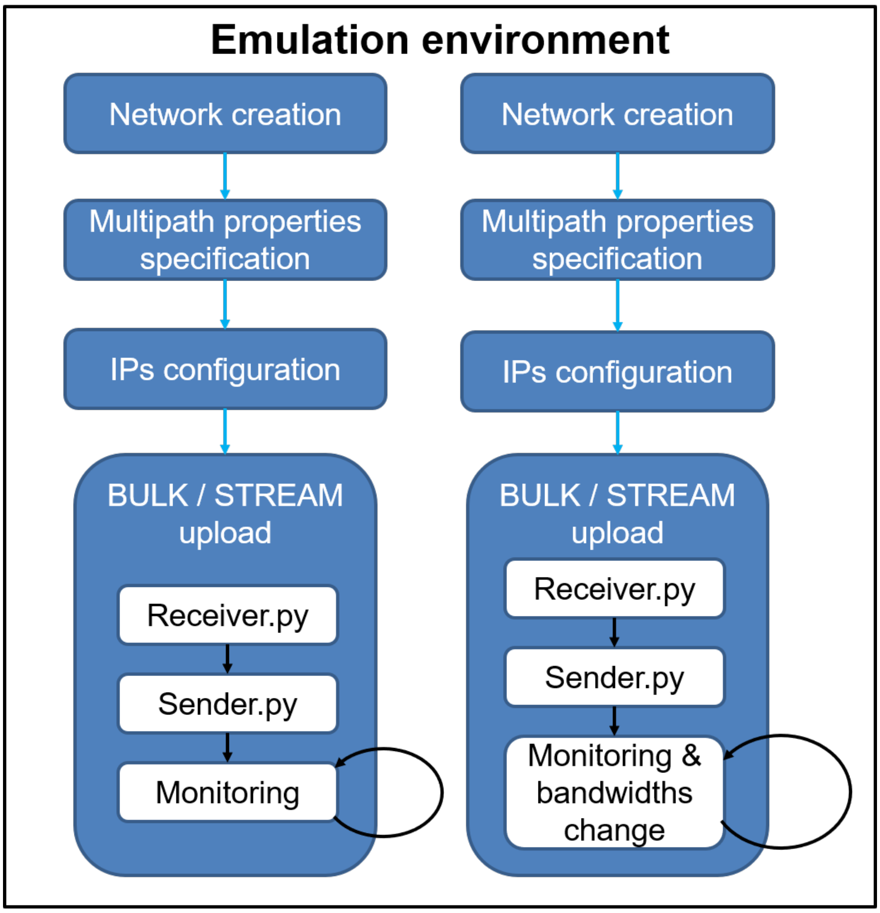


Figure 5.2: Network emulation flow for static (left) and dynamic (right) bandwidths. The network creation includes the process to define the MiniNet network topology. After configuring the main parameters for a multipath transmission, the IPs of the sender and receiver are set up. In addition to monitoring the communication between sender and receiver, in a dynamic environment (right) the available bandwidth is modified every second according to the current trace.

client-server procedure in Figure 4.2 is reconstructed by executing the receiver before the sender script. The transmission is monitored for the whole simulation time exploiting the Bandwidth Monitor NG [44].
The simulation flow of a dynamic scenario, on the right side of Figure 5.2, is structured to being modular and nearly equivalent to the static case. In particular, there is one major difference that consists of an additional step in parallel to the monitoring one. It permits to change at runtime the bandwidths of the desired interface of a node in the network topology with a smooth change. It is achieved by using the *config* function on an existing node as documented in [45].

5.4 Definition of Evaluation Experiments

Figure 5.3 schematizes the tree graph of the performed experiments to analyze the new MP-DPA and compare it to the off-the-shelf MPTCP scheduler and the standard TCP.

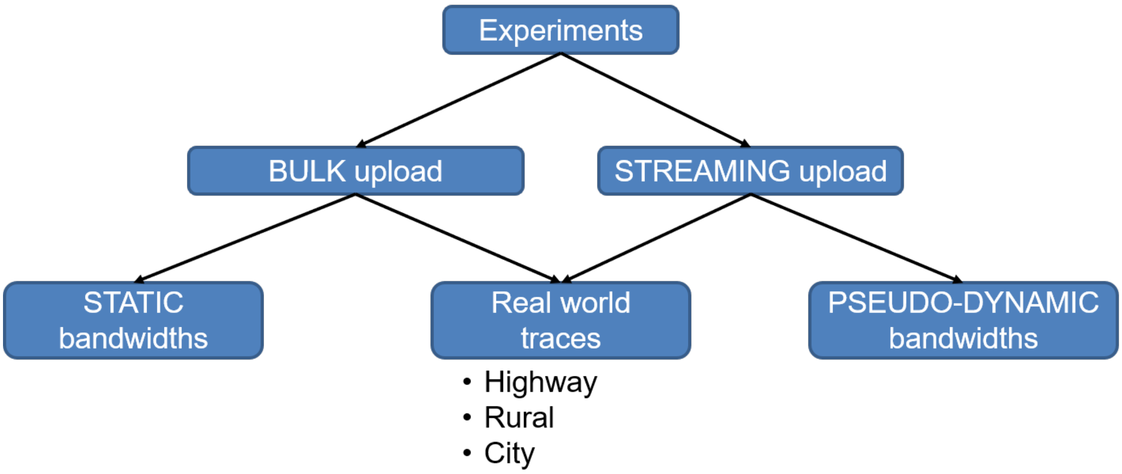


Figure 5.3: Followed approach for the characterization of the experiments. MP-DPA performances are analyzed for vehicle data/bulk and streaming upload. In both cases, the conducted experiments start from simple tests such as static or pseudo-dynamic scenarios to dynamic real-world traces in highway, rural, and city locations.

Starting from the left, the first experiment is characterized by bulk upload of data in static conditions. For bulk upload is intended the procedure for sending fixed-size blocks of data that are sent one at a time from the client (vehicle) to the server (cloud). For instance, a car continuously gathers information from the surrounding environment (e.g. remote diagnostic and location intelligence) and after a limited amount of time, the acquired data are sent to the cloud through a communication protocol.

Proceeding to the middle section of Figure 5.3, instead of having fixed bandwidths as in the static scenario, MP-DPA is simulated with a trace-driven approach. Here, multiple traces are obtained from real measurements and successively assigned to both sender interfaces. The dynamic simulations will focus on 3 different locations: highway, city, and rural that represent essentially the possible scenarios in the automotive context.

Since MP-DPA is based on MP-DASH [1] that is just validated in bulk conditions, this thesis investigates if MP-DPA can be also used for uploading a continuous stream of data as it happens for video streaming cases. From the simulation point of view, packets are continuously sent over time without interrupting the transmission. In this case, instead of analyzing MP-DPA using a static approach, it is tested in controlled pseudo-dynamic conditions, in addition to the dynamic ones.

Everything is preceded by a simple validation test that compares the main metrics to MP-DASH [1], the literature framework on which MP-DPA is inspired to.

5.5 Experiments Parametrization

According to Figure 5.3, MP-DPA is simulated in multiple conditions to provide a full characterization of its strengths and weaknesses. For each test, the parameter configuration depends on the type of experiment and the network conditions. Hence, in the following sections are reported the motivations behind the choice of such parameters for the simulation of every test.

5.5.1 Bulk Upload in Static Conditions

In the first experiment, the sender uploads 60 MB of data to the receiver in a bulk fashion within a static network. For this purpose, two different bandwidth types are taken into consideration: symmetric and asymmetric. Table 5.2 schematizes the properties of the sender interfaces in static conditions when a multipath protocol is enabled.
Both experiments are evaluated considering firstly a standard value of $\alpha = 1$ which is successively decreased to 0.8. Since 60 MB of data has to be sent in a bulk transmission, they are divided into

			Bandwidth configuration	
Name	**IP**	**Technology**	**Symmetric**	**Asymmetric**
Preferred interface	10.1.1.10	4G LTE	8 Mbps	4 Mbps
Non-preferred interface	10.1.2.10	4G LTE	8 Mbps	12 Mbps

Table 5.2: Definition of the configurations for the sender interfaces during static experiments and bulk upload.

6 packets of 10 MB with the constraint that each one has to be sent every 10 seconds.
Under this setup, using symmetric bandwidths, the sender takes 10 seconds to upload a packet using only the preferred LTE alone and 5 seconds when using standard MPTCP.
The asymmetric scenario instead, is configured in such a way to put MP-DPA in the worst-case conditions. Considering a bandwidth of 4 Mbps, the first interface will not be able to sustain alone the transmission since the ideal average time needed to upload one packet is 20 seconds. Thus, MP-DPA is forced to exploit the non-preferred interface to comply with the deadlines but this means achieving fewer data savings and thus more expensive upload.
As can be noticed, although symmetric and asymmetric are characterized by different speeds, the total available bandwidth is configured to the same value of 16 Mbps. In this way, every 10 seconds a packet is scheduled to be transmitted, and, using MPTCP, it can be ideally sent in not less than 5 seconds. The 3 proposed deadlines in which MP-DPA will be evaluated are chosen with multiple levels of difficulties:

- Deadline equal to the sending frequency - 10 seconds

- Medium deadline - 9 seconds

- Strict deadline - 7.5 seconds

In 6.4, lower deadlines are taken into consideration to understand which are the limits of the multipath protocols.

5.5.2 From Vehicle Sensor Data Upload To Streaming Data Upload

MP-DPA is implemented to provide advantages and optimization to the standard multipath protocols in an automotive environment. For this reason, it is firstly in-depth evaluated and validated in a bulk transmission typical of a car-to-cloud communication. Moreover, since MP-DPA is based on MP-DASH [1] that is just validated in bulk conditions, this thesis investigates MP-DPA performance in a completely new scenario such as the streaming one.
Practically speaking, during a bulk transmission, firstly, the vehicle maintains idle the network interfaces while acquiring the data through its sensors. After a while, the data collected until this moment are sent in a bulk fashion.
Contrarily, in a streaming transmission, the idle time proper of bulk upload must not be present, but the vehicle sends the data as soon as they are acquired. From a simulation point of view, this behavior is replicated by trying to reduce considerably the time between the transmission of two consecutive packets.

Streaming Upload in Static Conditions

In the first experiment, the sender uploads 60 MB of data to the receiver as the static bulk scenario, but in a stream fashion within a pseudo-dynamic network. Since MP-DPA is deadline aware protocol, the introduction of packets is required also for a stream-like scenario. Thus, the payload is divided into 60 packets of 1 MB, each one transmitted every second.
For pseudo-dynamic network is intended a heterogeneous set of bandwidths which are attributed to each interface as reported in Table 5.3. The preferred interface bandwidth is configured to

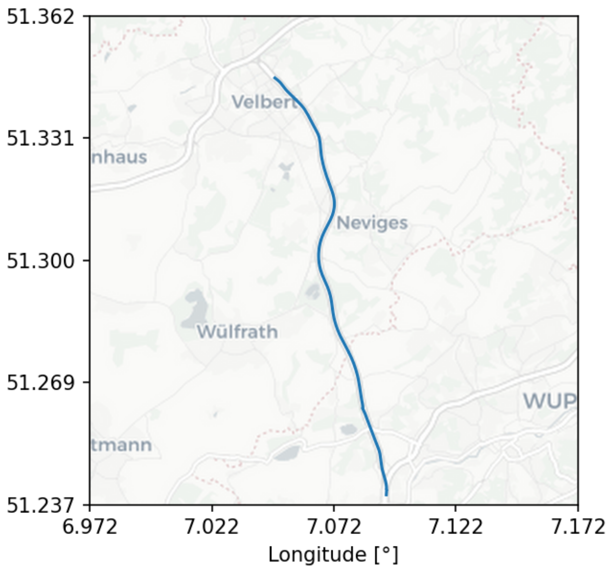
Name	IP	Technology	Bandwidth
Preferred interface	10.1.1.10	4G LTE	Triangular shape with peak of 14 Mbps
Non-preferred interface	10.1.2.10	4G LTE	6 Mbps

Table 5.3: Definition of the configurations for the sender interfaces during pseudo-dynamic experiments and streaming upload.

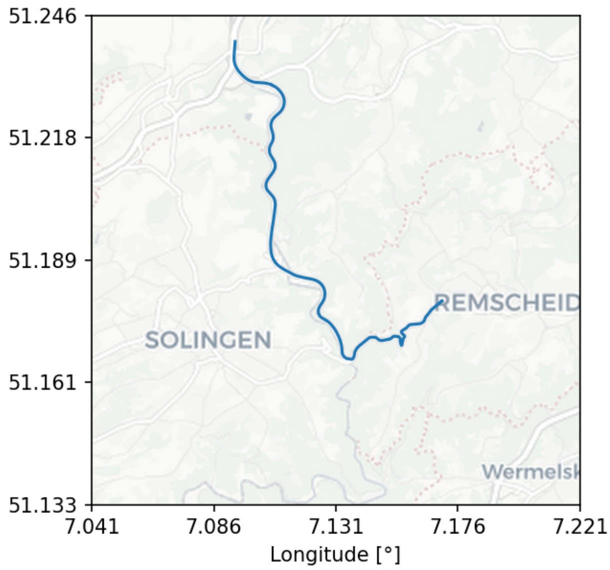
assume a triangular shape with a high of 14 Mbps while the non-preferred one is fixed to 6 Mbps. These particular choices of settings are motivated by the fact that in this way, it will be easier to understand if MP-DPA still works properly. In fact, according to the MP-DPA principle, the second interface must be enabled only when the total throughput is lower than the one required to send the entire packet. In this case, since 1 MB of data is completely sent in 1 second by using at least a bandwidth of 8 Mbps, the non-preferred interface must be enabled only at the proximity of the triangular shape minimum.
The proposed deadlines in which MP-DPA will be evaluated started from a relaxed value of 1 second to 0.9 and then 0.8 seconds.

5.5.3 Trace-Driven Experiments

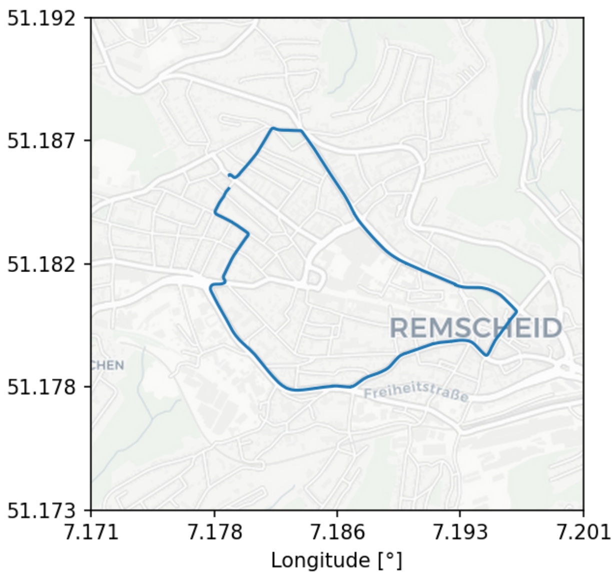
The feature that characterized all the static experiments was that both bandwidths were manually set to a fixed and known value. Unfortunately, in the real world the data rate are not stable and the coverage of network providers is not uniform. For this reason, MP-DPA in investigated under



(a) Highway scenario - A535.



(b) Rural scenario - District of Kohlfurth (Wuppertal)



(c) City scenario - City of Remscheid

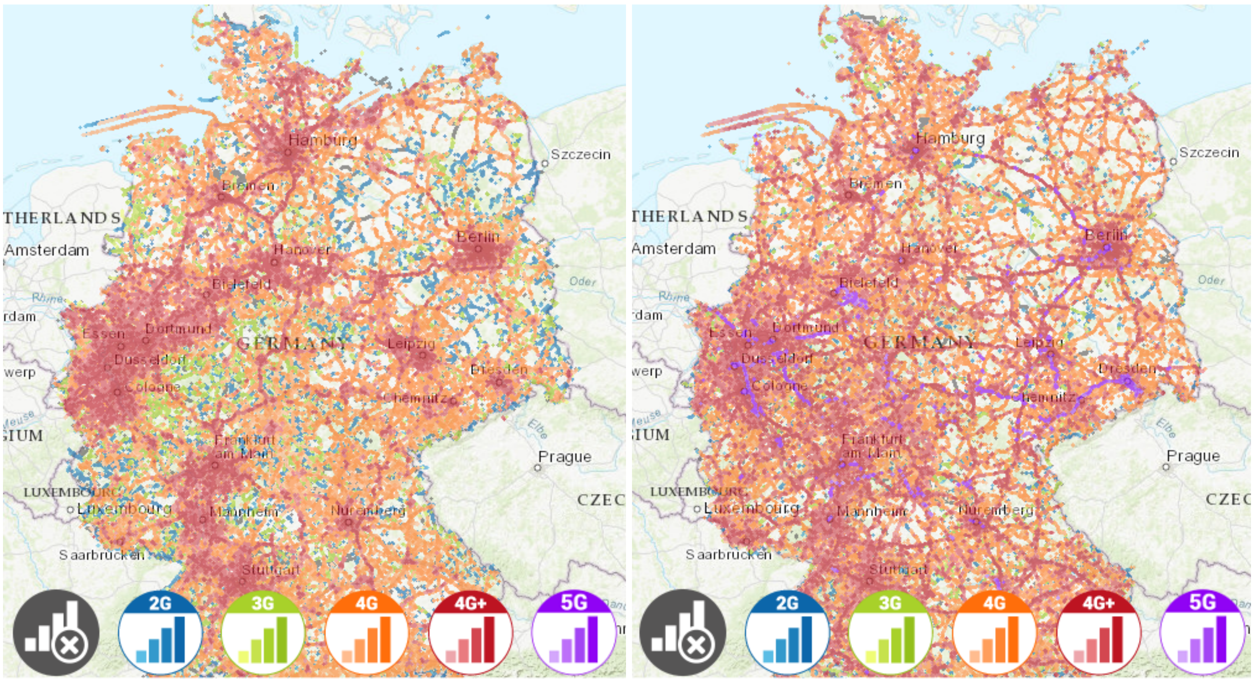
Figure 5.4: Routes followed by car to acquire the measurements for the dynamic experiments.

realistic settings through extensive in-field experiments to understand if its estimation principle can be compromised by the variability of the bandwidth.
In particular, the experiments are conducted at different locations in the North Rhine-Westphalia state in highway, rural, and city environments.

Firstly, the highway A535 that connects Wuppertal and Velbert cities is taken into consideration. For the second experiment, the traces are taken from a driving test in the rural district of Kohlfurth in Wuppertal. As rural is intended a particular geographic area that is located outside towns and cities where is likely that the network coverage is compromised somehow. In the third and last scenario, all the measurements are taken from the driving test in the city of Remscheid in North Rhine-Westphalia state.

Figure 5.4 shows the pre-defined routes that are followed during the experiments, while the video of rural and highway measurements is available at [46].

During the car-driving experiments, the throughput is computed through a simple *iperf3* test using the LTE provided by two commercial carriers in Germany: O2 (Telefonica) and Telekom. According to [47], O2 is characterized by an overall lower cost with respect to Telekom. At expense of this, Telekom is able to reach a more uniform network coverage and higher speeds as Figure 5.5 highlights.



(a) O2 (Telefonica) **(b)** Telekom

Figure 5.5: Connection speed and network coverage in Germany for different German providers. In addition to a more uniform network coverage, Telekom **(b)** can reach higher speeds (5G) than O2 **(a)**. Although O2 lacks in performances, it is considered cheaper than Telekom [48]. Data provided by [47]

.

During the MP-DPA and MPTCP simulations, the preferred interface is associated with O2 measurements while Telekom is represented by the non-preferred one. The main goal of MP-DPA remains the same: minimize the usage of Telekom and thus the overall transmission cost, without compromising the reliability of MPTCP.

The experiments are structured similarly to the static scenario. For each location, MPTCP and

MP-DPA are compared in terms of throughput behavior, data distribution, and deadlines missed. Using bandwidths taken from real driven tests, the bandwidths can reach higher values than the static experiment. Thus, for every simulation, the transmission is firstly configured as bulk with packets of 100 MB that are sent every 60 seconds. Since the connection is not stable, it is quite common that a packet is not sent completely unless the connection has a sufficient speed during the whole sending time.
As static experiments, also here, the 3 proposed deadlines in which MP-DPA will be evaluated have multiple levels of difficulties:

- Deadline equal to the sending frequency - 60 seconds
- Medium deadline - 50 seconds
- Strict deadline - 40 second

During the streaming upload instead, the transmission is configured with smaller packets of 2 MB that are sent every second. The packet length and sending frequency are computed to maintain similar proportions with the ones adopted in bulk experiments.
The 3 proposed deadlines in which MP-DPA will be evaluated have multiple levels of difficulties:

- Deadline equal to the sending frequency - 1 second
- Medium deadline - 0.9 seconds
- Strict deadline - 0.8 seconds

6 Experimental Results

In the following section, the experimental validation takes place. Multipath Deadline and Preference Aware (MP-DPA) is compared to standard communication network protocols such as TCP and MPTCP. The new framework performances are measured in static conditions with the investigation of possible pareto-optimal values for the deadline sensitivity. In the last part of the chapter, the focus is shifted on the results of real traces experiments obtained by drive test measurements in different locations.

6.1 Validation and Comparison to Literature Results

Before proceeding with the experiment results, the basic features of the new Multipath Deadline and Preference Aware (MP-DPA) algorithm are compared to an already existing scheduler called Multi Path Dynamic Adaptive Streaming over HTTP (MP-DASH) [1] which MP-DPA is inspired to. The scheduler implemented by Han *et al.* is a multipath deadline-aware scheduler designed

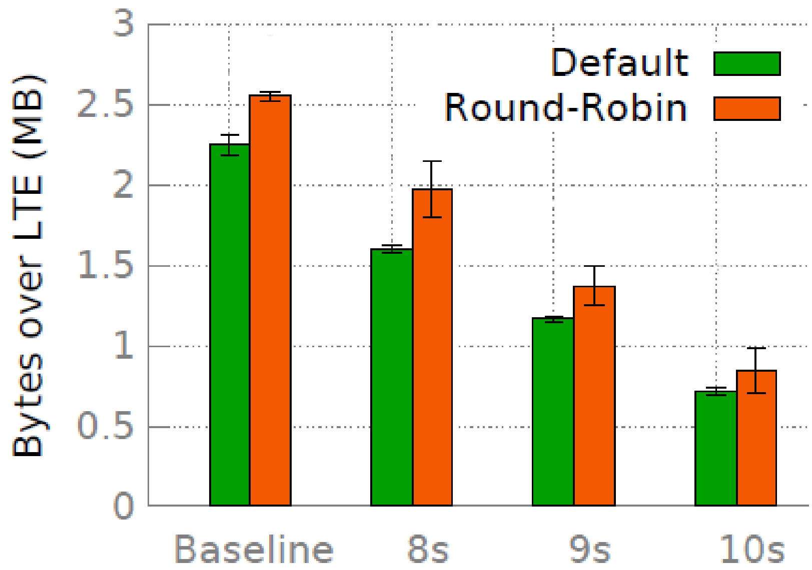


Figure 6.1: Data sent over the second LTE interface using the literature framework MP-DASH [1]. Baseline MPTCP with default scheduler sends approximately half of the data over LTE (2.2 MB). MP-DASH can reduce this value till ∼0.8 using a deadline of 10 seconds. With stricter deadlines, the savings are less considerable but still present.

for video streaming which aims to reduce cellular data usage and maximize the use of WiFi.
The test consists of a simple scenario where the client uploads from server 5 MB of data. The bandwidth of interface 1 and 2 (WiFi and cellular for MP-DASH and preferred and non-preferred

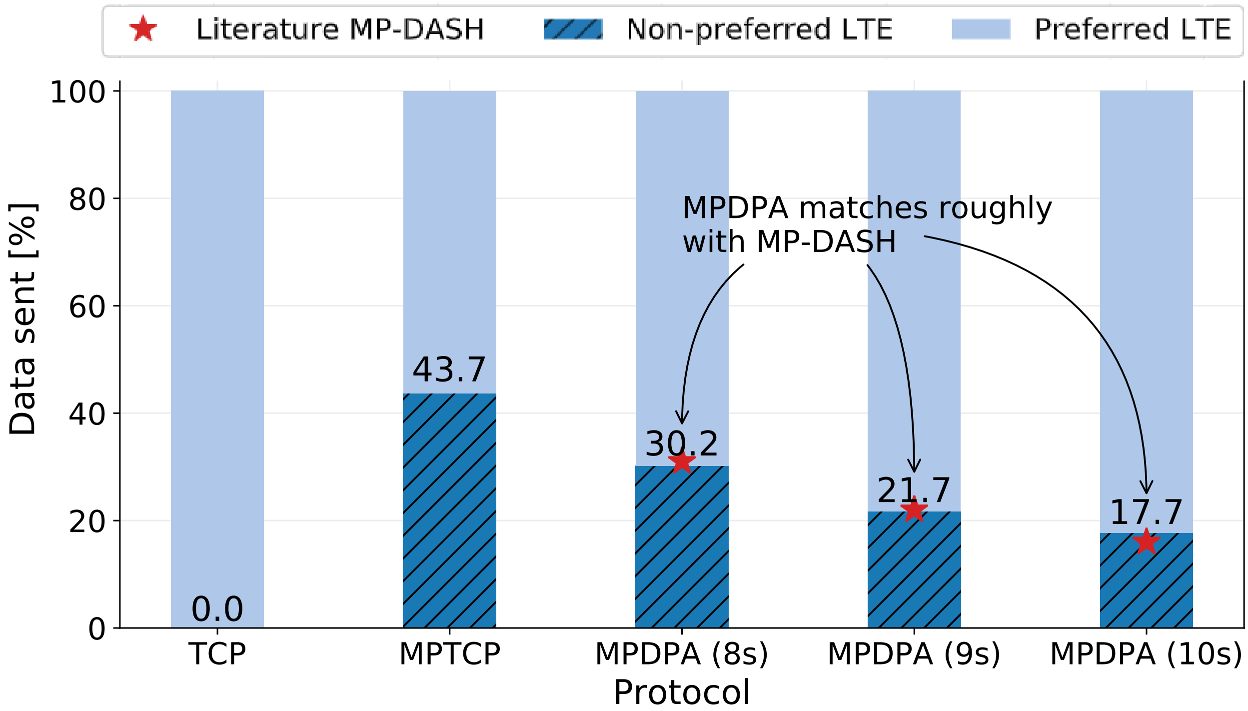


Figure 6.2: Data distribution of the Multipath Deadline and Preference Aware (MP-DPA). TCP does not exploit the seconds interface while MPTCP, again, divides equally the data among the interfaces. MP-DPA reduces to 17.7% the data consumption of the non-preferred interface for 10 seconds of deadline, which correspond to 0.9 MB, a similar value to the one obtained in Figure 6.1. The savings for the 9 and 8 seconds of deadline are lower but again comparable to MP-DASH.

interface for MP-DPA) are throttled to be 3.8 and 3.0 Mbps, respectively. Under this setup, the sender takes 10.5 seconds to upload the 5 MB using interface 1 alone and 6 seconds when using MPTCP.

Despite the document of Han examines three metrics using default and round-robin scheduler: download time, cellular bytes, and radio energy consumption, only the first two with default scheduler are taken into consideration to validate MP-DPA.

In Figure 6.1 is plotted the number of bytes over the non-preferred interface obtained using MP-DASH for a total upload of 5 MB. In Figure 6.2, are shown the same percentages but using the new MP-DPA. In both cases, the results are compared for different deadlines.

From a global point of view, in both cases, the second interface data usage is significantly reduced compared to baseline MPTCP. MPTCP tries to use always both interfaces (half data sent over the first interface and the other half over the second interface), while MP-DASH and the new MP-DPA cut down considerably the usage of the non-preferred one. In particular, the longer the deadline is, the more the savings are. As we wanted to demonstrate, the obtained results are very similar. For instance, when the deadline is 10 seconds, in both cases only ∼18% of data (that corresponds to ∼0.9 MB) are sent over the non-preferred interface.

For both experiments, each packet is entirely transmitted before the deadline expires. However, under non-ideal scenarios with low overall bandwidths, such as the asymmetric one in 6.3, the deadline may be missed. As we anticipate before, missing a deadline can further be prevented by tuning the parameter α in Algorithm 1. Since in the previous experiment the percentage of missed deadlines is 0, α is set to a standard value of 1. For other measurements, MP-DPA will be experimented also with different α values.

6.2 Favorable Conditions: Bulk Upload with Symmetric Bandwidths

The symmetric scenario can be defined as the basic and most favorable one. The first and second bandwidths are set to 8 Mbps, a value that ideally should be enough to send 10 MB every 10 seconds using just one interface. Simulating the network for 60 seconds, however, multiple metrics can be evaluated depending on the used protocol, the deadline, and the deadline sensitivity α for MP-DPA.

6.2.1 Basic Performances Assessment

Figure 6.3 presents the effective throughput for different protocols. Proceeding from top to bottom, the standard TCP, being a single path protocol, uses only the preferred interface at maximum speed.

Moving to MPTCP (middle), both interfaces are fully exploited researching the maximum performances. Every deadline is largely respected since, for approximately half of the interval, both interfaces are idle. At the expense of this, the non-preferred interface is exploited as the preferred one. The bottom part of both graphs, show the solution proposed by MP-DPA. For a relaxed deadline of 10 seconds, contrarily to the preferred interface that remains fully exploited for more time, the non-preferred one is enabled only when needed. Each peak represents the moment in which the sender recognizes that the first interface is no more able to sustain alone the transmission of a packet (Algorithm 1). In this way, the second interface is enabled or disabled depending on the goodput estimation in a given time.

The last plot of Figure 6.3 displays again MP-DPA with a configured deadline of 7.5 seconds. TCP and MPTCP are not displayed for a lower deadline since they are not deadline-aware protocols. For them each interface is exploited in the same way for every deadline, resulting in similar behaviors. MP-DPA instead, shows different results. The non-preferred interface peaks "remain high" for more time because if the same amount of data has to be sent in less time, the total throughput must be higher.

To have a better idea of how the throughout is distributed among the two LTE interfaces, in Figure 6.4, Figure 6.5 and Figure 6.6, are plotted the probability density functions of MPTCP and MP-DPA. From Figure 6.4 can be concluded that the usage of the interfaces are almost symmetric to each other if MPTCP is used. Here, no distinction between preferred and non-preferred interface is made. Figure 6.5 clearly shows that MP-DPA focuses on fully exploiting the preferred interface since its throughput values are concentrated at 8 Mbps. The non-preferred LTE speeds are concentrated in the first half of the graph meaning that a low amount of data is sent over this interface. Using a lower deadline, Figure 6.6 displays that the non-preferred LTE peak is higher and the overall distribution tends to the MPTCP one.

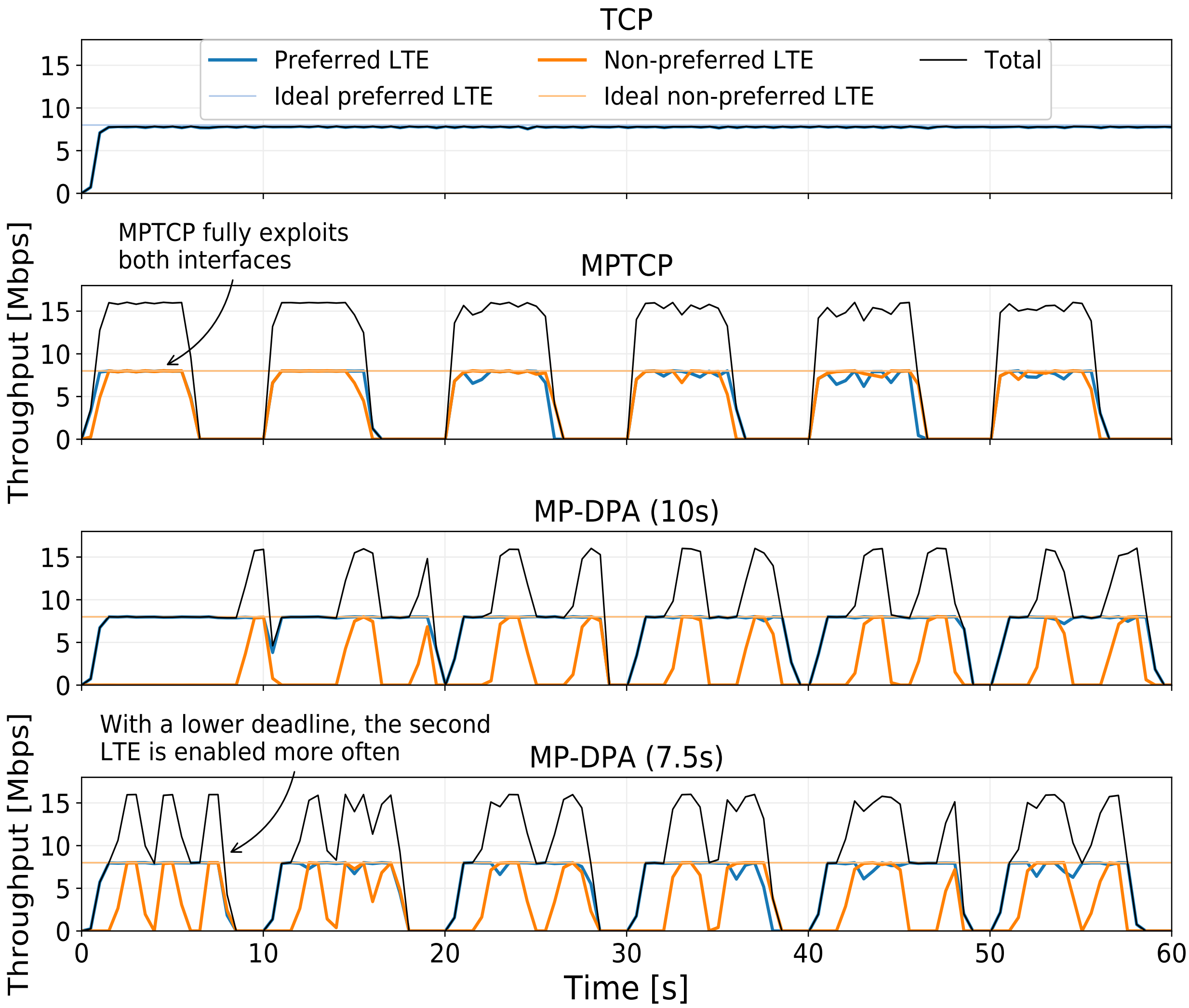


Figure 6.3: Effective throughput during a bulk transmission of 6 packets in symmetric conditions using different frameworks. TCP consistently uses the first interface, while MPTCP fully exploits both interfaces researching always the maximum performances. With a deadline of 10 seconds, MP-DPA enables the non-preferred interface to meet the deadline. By lowering the deadline to 7.5 seconds, MP-DPA activates more often the non-preferred interface to increase the transmission speed.

Till this moment, just the throughput behavior is analyzed. In the case of MP-DPA, since it depends on the deadline, how the overall latency is influenced by it? The answer can be observed in Figure 6.7. It represents the average time needed to send a packet of data for every protocol depending on the deadline value.

With TCP, each packet needs on average ∼10 seconds to be entirely sent. Thus, the deadline is rarely respected for a single-link connection. Using a multilink protocol instead, the time is reduced considerably. Independently from the deadline, MPTCP researches always the maximum performances and the time to send 10 MB remains stable at 5.2 seconds. MP-DPA shows its typical adaptive behavior. The latency is directly proportional to the deadline and the deadline sensitivity α. The lower the deadline and α are and the faster the packet is sent, as suggested by the more

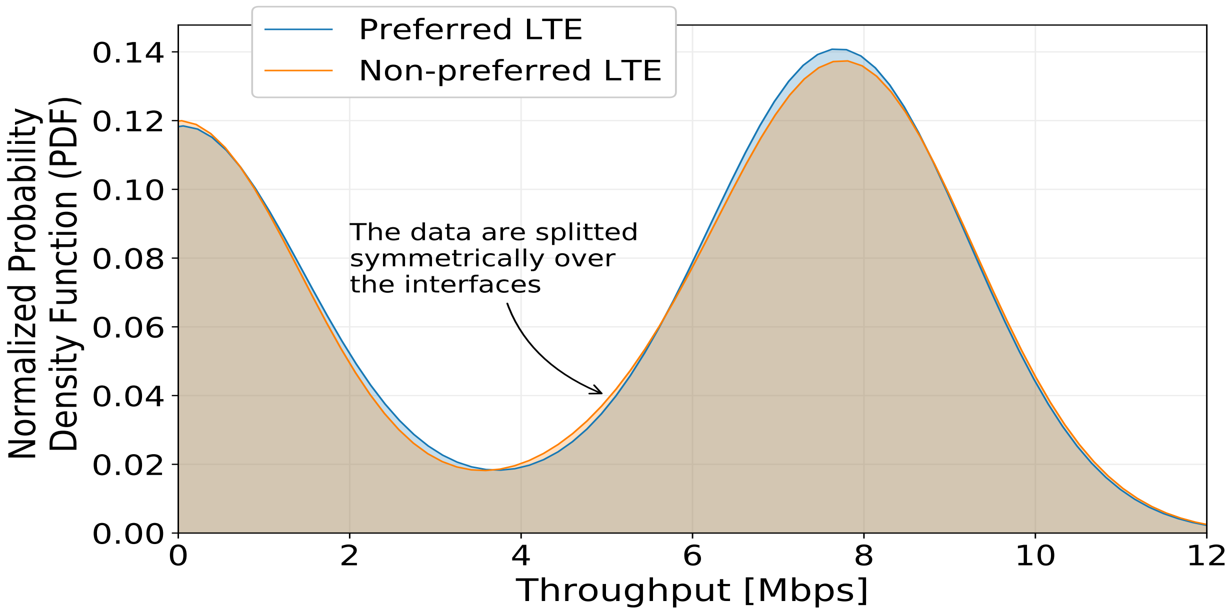


Figure 6.4: Multipath Transmission Control Protocol (MPTCP) shows a symmetric probability density function independently from the deadline value. There is no distinction between preferred and non-preferred LTE.

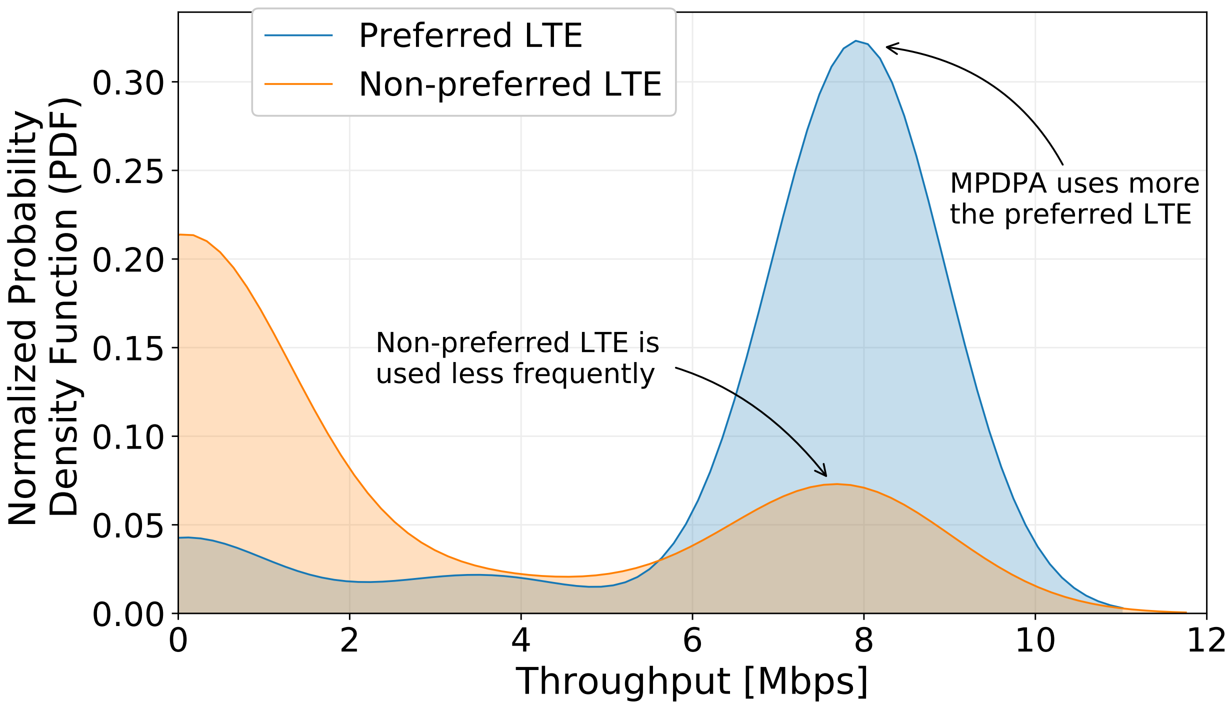


Figure 6.5: PDF of MP-DPA with deadline = 10 seconds. MP-DPA maximizes the usage of the preferred interface while reducing the non-preferred one. The preferred LTE is used more than MPTCP Figure 6.4.

frequent occurrence of peaks in the last plot of Figure 6.3 with a lower deadline.

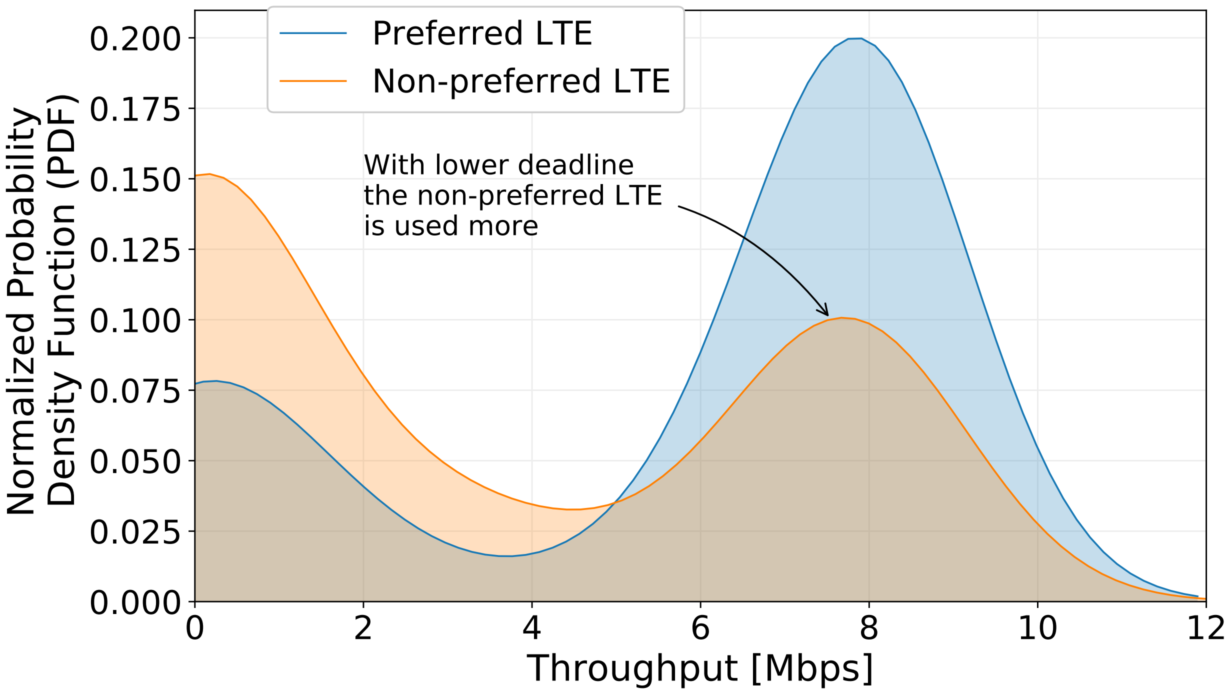


Figure 6.6: With a deadline of 7.5 seconds, MP-DPA exploits more often the non-preferred interface. It results in a distribution of the seconds LTE more shifted toward higher throughput values.

6.2.2 Non-Preferred Interface Usage and Reliability Measurements

Till now, all the considerations introduce and explain how MP-DPA adapts its communication performances depending on the deadline value. However, the new MP-DPA is designed as a deadline and preferences-aware protocol to reduce the non-preferred interface usage while respecting deadlines. To test whether MP-DPA meets its specifications, in the following are analyzed the data sent on both interfaces and the number of deadlines missed.

Figure 6.8 shows the percentages of data distribution among the preferred and non-preferred interfaces of TCP, MPTCP, and MP-DPA for different deadlines. The highlighted part of the bars is the most interesting metric from the MP-DPA point of view and represents the percentages of data sent over the non-preferred LTE.

MPTCP shows again a symmetric behavior: in all 3 deadlines, since the bandwidths are equals, the data are sent equally over both interfaces. MP-DPA cuts down considerably the usage of the non-preferred LTE: $\sim$25% of savings for a deadline of 10 seconds that decreases to $\sim$13% with a lower deadline.

The reliability of each protocol as the percentage of deadline missed is plotted in Figure 6.9. Since the overall reachable throughput using a multipath protocol is enough to sustain all the packets transmission, for MPTCP and MP-DPA all the deadlines are respected independently from the value of α. However, the same result is not achieved by TCP which shows very high percentages of missed deadlines.

By decreasing the value of α to 0.8, since the virtual deadline seen by the sender is lower than the

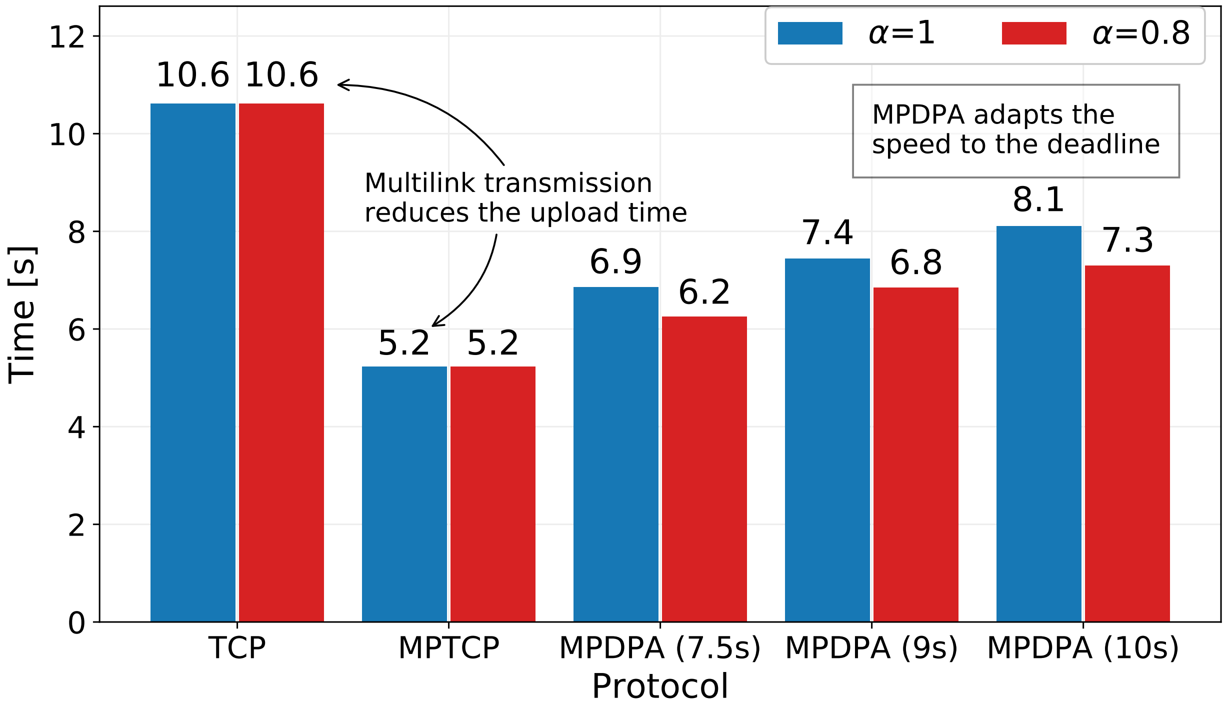


Figure 6.7: Average time needed to send a packet of 10 MB with TCP, MPTCP, and MP-DPA with different deadlines for a bulk upload in symmetric conditions. A multilink transmission reduces considerably the upload time. MP-DPA can adapt its speed resulting in a trade-off between speed and savings. By lowering the value of α, more often the non-preferred interface is enabled, resulting in a faster transmission. α does not influence TCP and MPTCP performances.

real one, it activates more often the second interface to increase the transmission speed. The red bars in the previous plots show the consequences of this change.

Also looking back at Figure 6.7, as the overall throughput increases, the time needed to send 10 MB of data with $\alpha = 0.8$ has a lower value to $\alpha = 1$. Consequently, Figure 6.8 demonstrates that there is a difference of data sent over the non-preferred interface from +5% to +9% between $\alpha =$ 0.8 and 1. However, in this experiment, a lower value of α does not influence the percentage of deadline missed that is fixed to 0 (Figure 6.9). Thus, a smaller value of α can be only useful to improve the reliability wherever it is not optimal, but it leads to worse results in terms of data minimization of the non-preferred LTE.

In an optimal network configuration such as the symmetric one, MP-DPA provides an MPTCP-like transmission characterized by high performances and reliability. Furthermore, it minimizes the non-preferred LTE usage as requested by the specifications. The next section deals with the evaluation of MP-DPA in non-favorable conditions proper of asymmetric bandwidths.

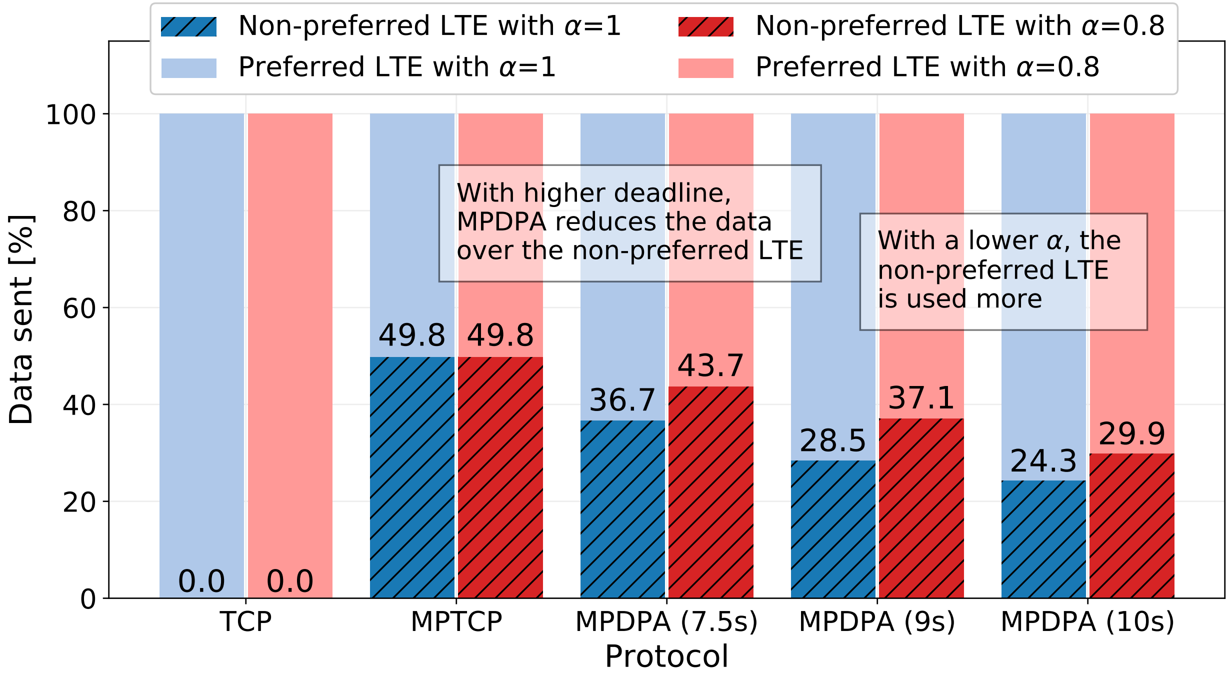


Figure 6.8: Data distribution for TCP, MPTCP, and MP-DPA with different deadlines during a bulk transmission of 60 MB of data in symmetric conditions. TCP does not use the second interface, while MPTCP divides symmetrically the data over the LTEs. MP-DPA savings decrease by reducing the deadline and α

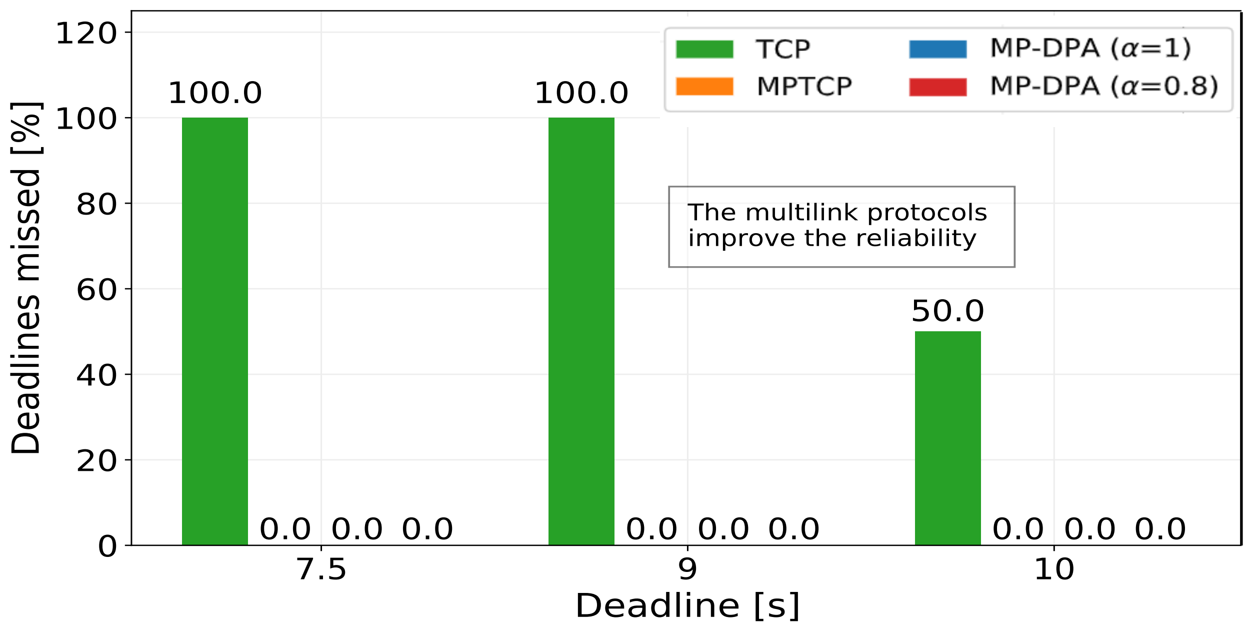


Figure 6.9: Reliability analysis in terms of deadlines missed using TCP, MPTCP, and MP-DPA with different deadlines. A total of 6 packets are sent with bulk upload in symmetric conditions. Except for TCP that is too slow to satisfy the deadlines, MPTCP and MP-DPA can sustain a reliable transmission.

6.3 Disadvantageous Conditions: Bulk Upload with Asymmetric Bandwidths

One possibility to challenge MP-DPA always in static conditions is to configure the network asymmetrically. Differently than before, preferred and non-preferred LTE bandwidths are set to 4 and 12 Mbps, respectively. In this way, considering a lower bandwidth of 4 Mbps, the first LTE will not be able to sustain alone the transmission of every packet without compromising the reliability.

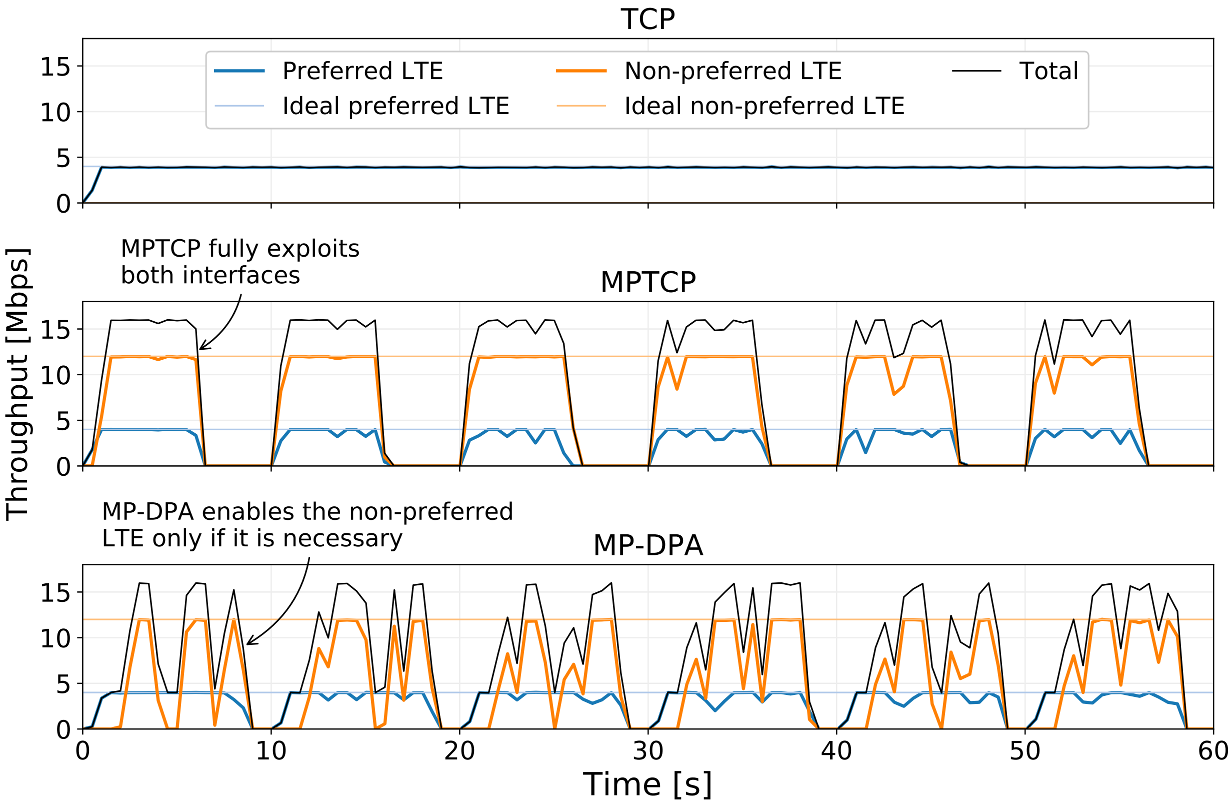


Figure 6.10: Effective throughput during a bulk transmission of 6 packets in asymmetric conditions using TCP (top), MPTCP (middle), and MP-DPA (bottom) with deadline = 10 seconds. Being the preferred interface too "slow" to sustain the transmission, MP-DPA is forced to use more often the non-preferred interface.

Thus, MP-DPA is put in conditions where it is forced to exploit the non-preferred interface to maintain the transmission reliable. This phenomenon is displayed very clearly in Figure 6.10 that represents the effective throughput of different protocols using a deadline of 10 seconds.
Comparing it with the symmetric scenario in Figure 6.3, all the protocols show a similar behavior although the bandwidths are different. In MPTCP, both interfaces are fully exploited during the whole transmission time, while MP-DPA presents the same peak-behavior produced by the enabling and disabling procedure of the non-preferred interface. However, since the orange area is bigger than the blue one, it can be noticed that more data are sent over the second subflow than the first

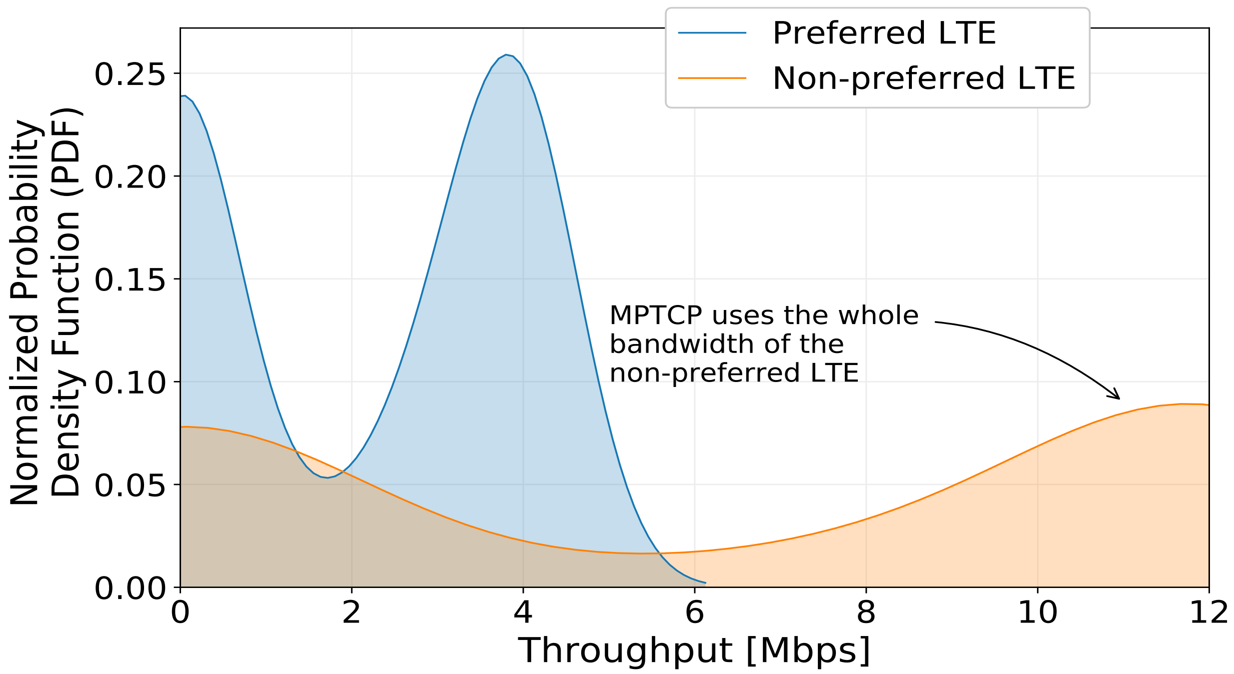


Figure 6.11: Probability density function of MPTCP during a bulk transmission in asymmetric conditions. MPTCP uses more likely the whole bandwidth of the non-preferred interface.

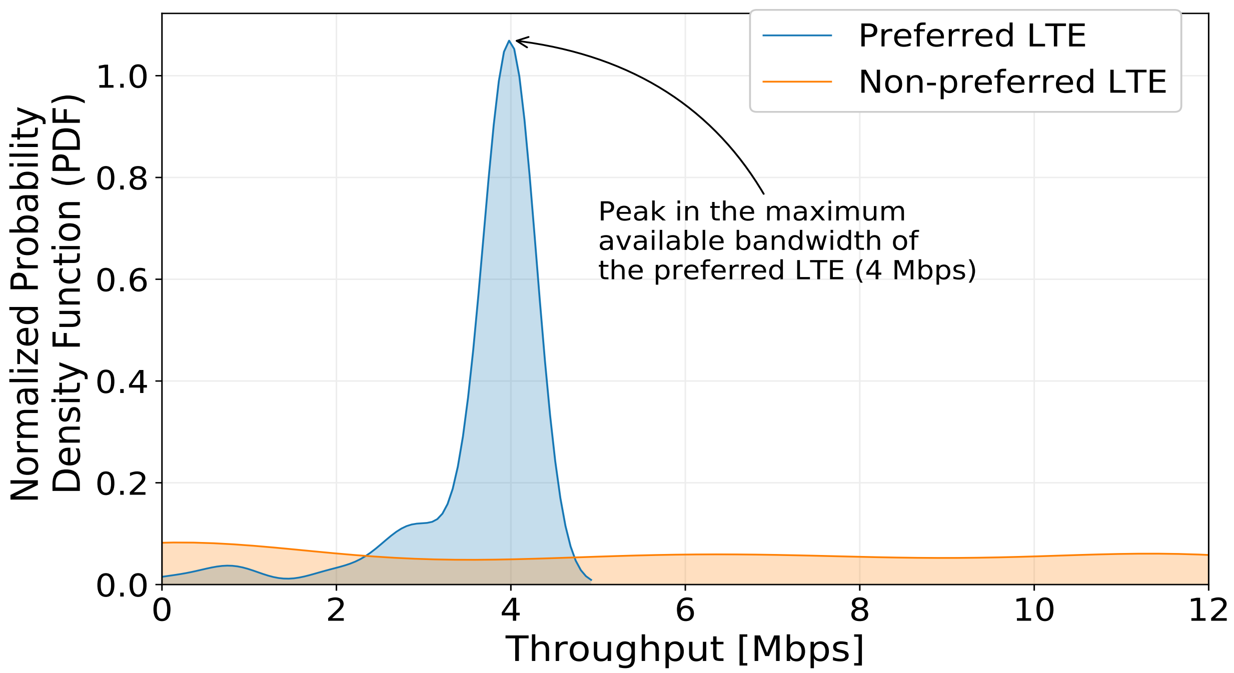


Figure 6.12: Probability density function of MP-DPA during a bulk transmission in asymmetric conditions. The preferred interface is fully exploited as the peak at its maximum available speed of 4 Mbps suggests. The usage of the non-preferred interface is still minimized.

one. From a probability distribution point of view, Figure 6.11 illustrates that MPTCP uses the non-preferred interface close to its maximum speed of 12 Mbps for most of the cases. In the case of MP-DPA in Figure 6.12, the non-preferred LTE throughput is spread among the whole speeds interval, while the preferred one is mostly used at full speed reaching the peak at the maximum exploitable bandwidth (4 Mbps).

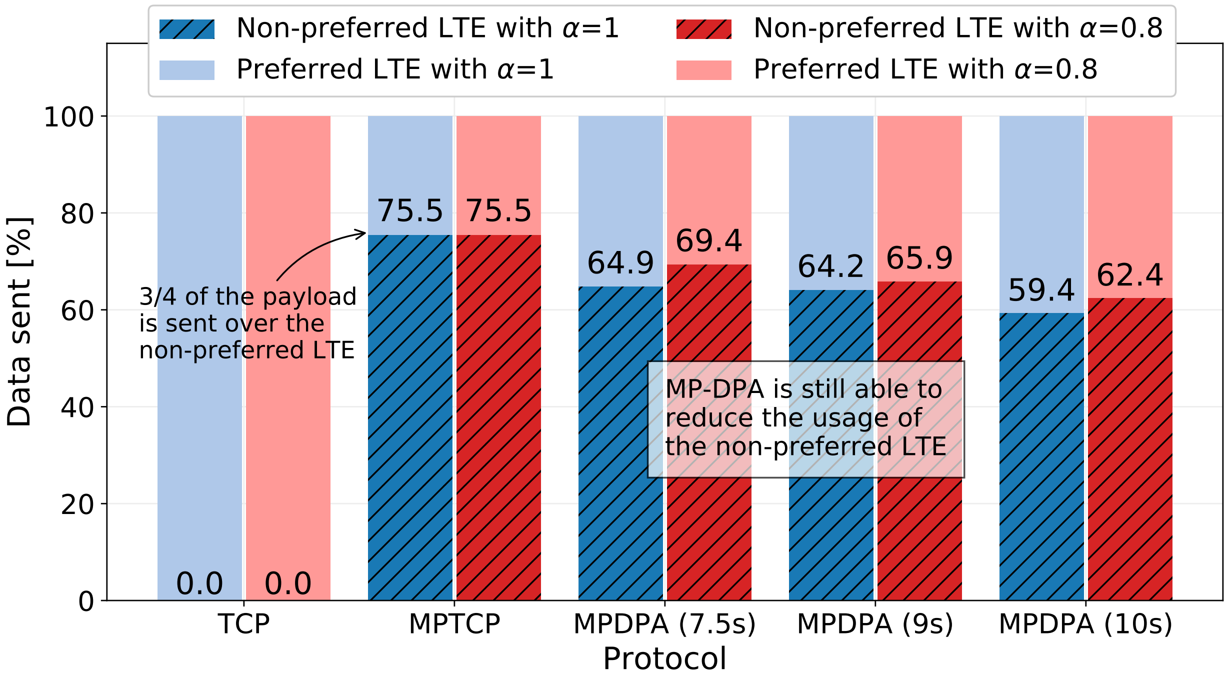


Figure 6.13: Data distribution for TCP, MPTCP, and MP-DPA with different deadlines during a bulk transmission of 60 MB of data in asymmetric conditions. MPTCP sent 3/4 of the data on the non-preferred interface independently from the deadline value. MP-DPA is still able to provide savings that decrease by reducing deadline and α.

As the symmetric scenario, also for the asymmetric one will be considered the MP-DPA metrics of interest such as the distribution of data sent among the interfaces and the percentage of deadline missed. In Figure 6.13, the highlighted area shows that in the case of MPTCP, 3/4 of the data are sent over the non-preferred interface. Although the network conditions are unfavorable for MP-DPA, a slight improvement can be seen for both α values. In the worst-case scenario with a deadline of 7.5 seconds, $\sim$10% fewer data are sent through the non-preferred LTE that increases to 16% for a higher deadline.

Although there are improvements in terms of savings, the percentages of missed deadlines are not tolerable as displayed in Figure 6.14. For strict deadlines, if MP-DPA is simulated with $\alpha = 1$, from 1 to 4 packets over 6 are not completely sent. Since MPTCP can sustain the whole transmission, the MP-DPA results are not satisfying.

To increase the reliability at the expense of more data sent through the non-preferred subflow, the α value is decreased to 0.8. The red bars of Figure 6.13 and Figure 6.14 show the obtained results. Comparing it to $\alpha = 1$, the amount of data sent over the non-preferred interface increases from 1.7% to 4.5% but all deadlines are respected. Thus, using $\alpha = 0.8$, MP-DPA equals the performances of

MPTCP and some savings of data sent the non-preferred interface are still present although smaller than standard MP-DPA with $\alpha = 1$.

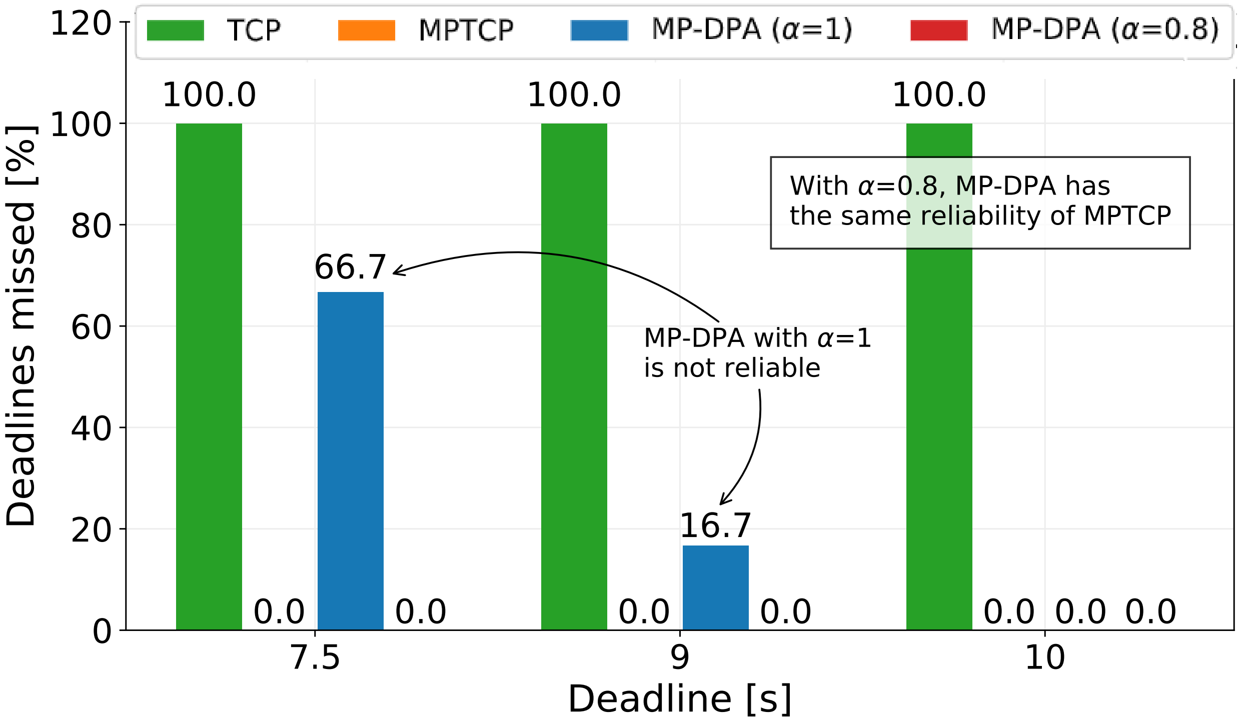


Figure 6.14: Comparing the deadlines missed using TCP, MPTCP, and MP-DPA with different deadlines. A total of 6 packets are sent with bulk upload in asymmetric conditions. Contrarily to MPTCP which can sustain a reliable transmission, MP-DPA with $\alpha = 1$ does not meet all the deadlines. By decreasing α, the reliability of MP-DPA is improved.

6.4 MPTCP and MP-DPA Limitations

From the results obtained in symmetric and asymmetric conditions, in both cases, MP-DPA enhances MPTCP in terms of usage of the non-preferred LTE. However, in all the previous experiments the most strict deadline is set to 7.5 seconds. Considering a total bandwidth of 16 Mbps, every 10 MB packet can be theoretically sent in 5 seconds, and thus, the deadline is always respected. How will MP-DPA and MPTCP practically behave if the deadline is set to be nearer and nearer to the limit of 5 seconds? The additional experiments in symmetric and asymmetric conditions conclude that multipath protocols, in practice, does not behave as expected theoretically.

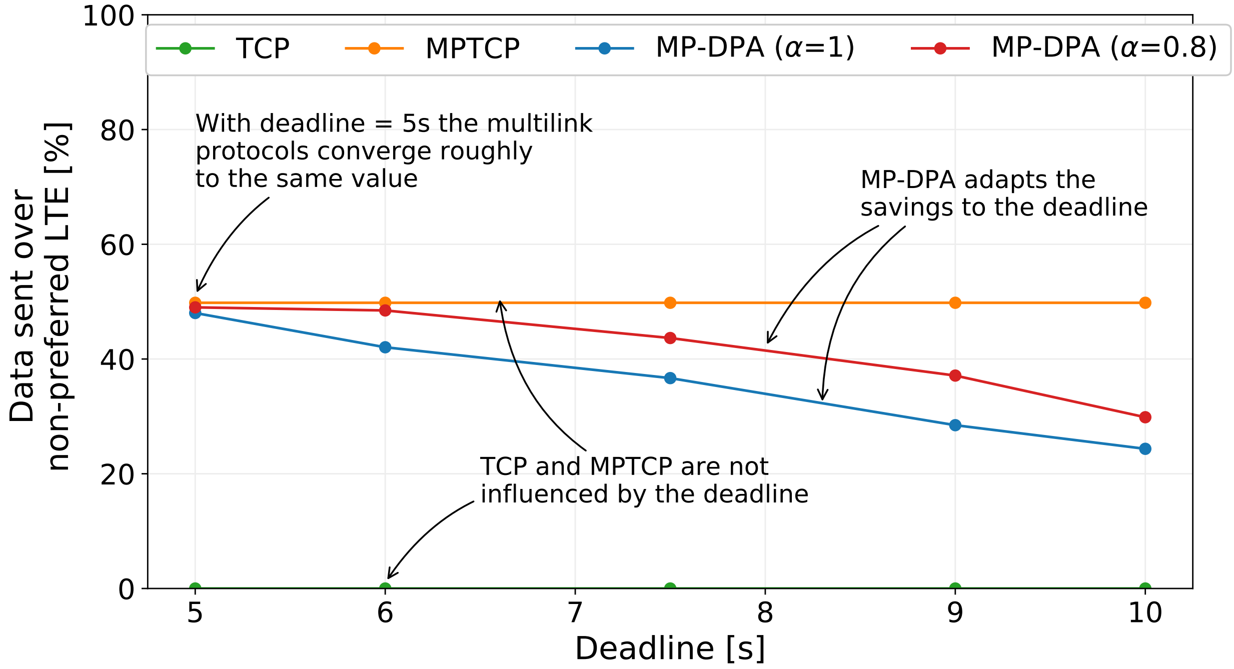
Figure 6.15 displays the resulting behavior for symmetric bandwidths. Starting from the data distribution in **(a)**, except for TCP that never uses the second interface and MPTCP that is stable to ∼50%, MP-DPA can adapt the interface usage to the deadline. Contrarily to MPTCP where the data is equally distributed independently from the deadline value, MP-DPA savings are directly proportional to the value of the deadline and α. With stricter deadlines, the savings decrease until they converge to the value of ∼50% as the baseline multipath protocol.

In **(b)** are reported the percentages of deadline missed for every communication protocol. As expected from the previous comparisons, the reliability decreases with a smaller deadline. Using α=1, for a deadline of 6 seconds, MP-DPA cannot anymore sustain the transmission and it is necessary to decrease α to 0.8 to obtain the same MPTCP performances. With a deadline of 5 seconds instead, all the protocols fail to send every packet. This means that even though both interfaces are fully exploited, the transmission results unreliable due to the non-ideality of the connection. A deadline of 5 seconds represents the limit for every multipath protocol and thus, also for MP-DPA.

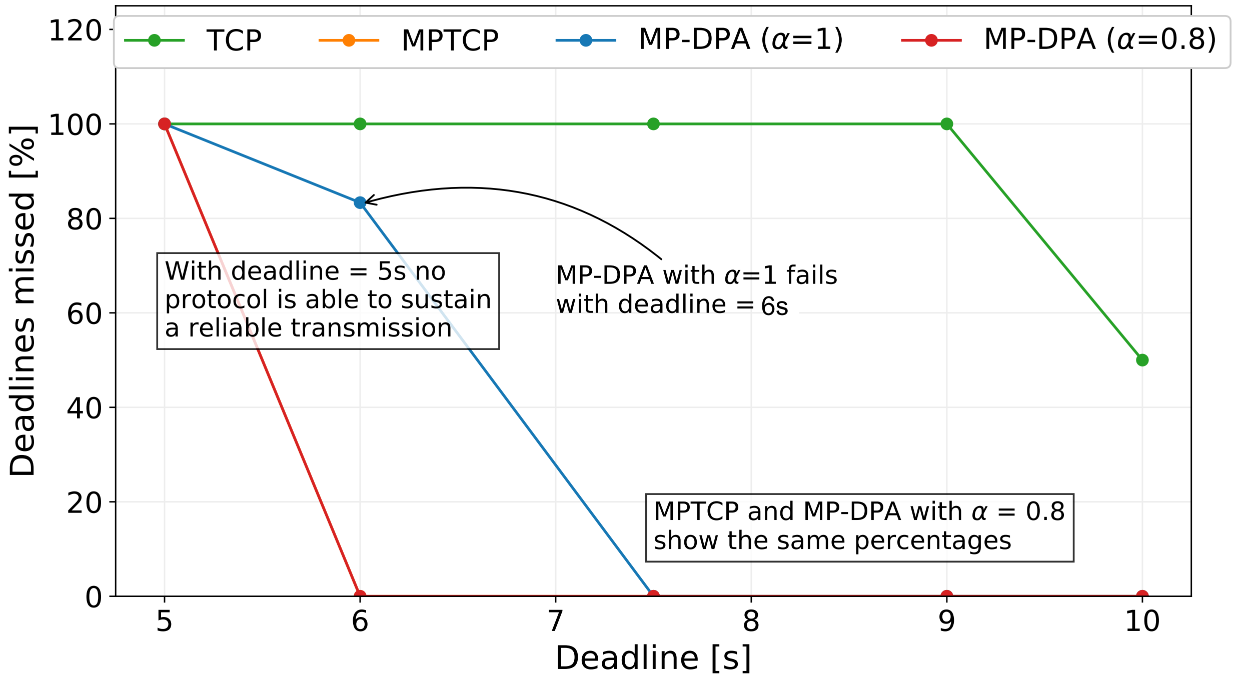
More interesting results can be observed in Figure 6.16 where MP-DPA is tested simulated using asymmetric bandwidths. Concerning the data sent through the non-preferred LTE, in Figure 6.16 **(a)**, the decreasing behavior of MP-DPA is repeated. Also here, by reducing α to 0.8, the second interface is used more often than $\alpha = 1$, but still less than MPTCP.

Being the asymmetric case more difficult to be managed for MP-DPA, slightly different percentages of deadline missed are reported in Figure 6.16 **(b)**. Starting from a deadline of 9 seconds, the standard MP-DPA is not able to send all the packets and it requires to lower α at 0.8. However, even though a lower deadline sensitivity is used, continuing to decrease deadline, at 6 seconds, 1 packet over 6 is not fully transmitted. This requires to decrease again α. Similarly to before, the limit of multipath protocols is defined for 5 seconds of deadline where every protocol does not complete the transmission of any packet.

Therefore, from these measurements, we can conclude that MP-DPA limitations, if the correct α is configured, are equals to MPTCP ones. Anyway, its performance depends heavily on the value of the deadline and α. To understand better this dependence, are conducted some experiments always in static conditions to investigate the existence of a possible pareto-optimal value of α in given network conditions.

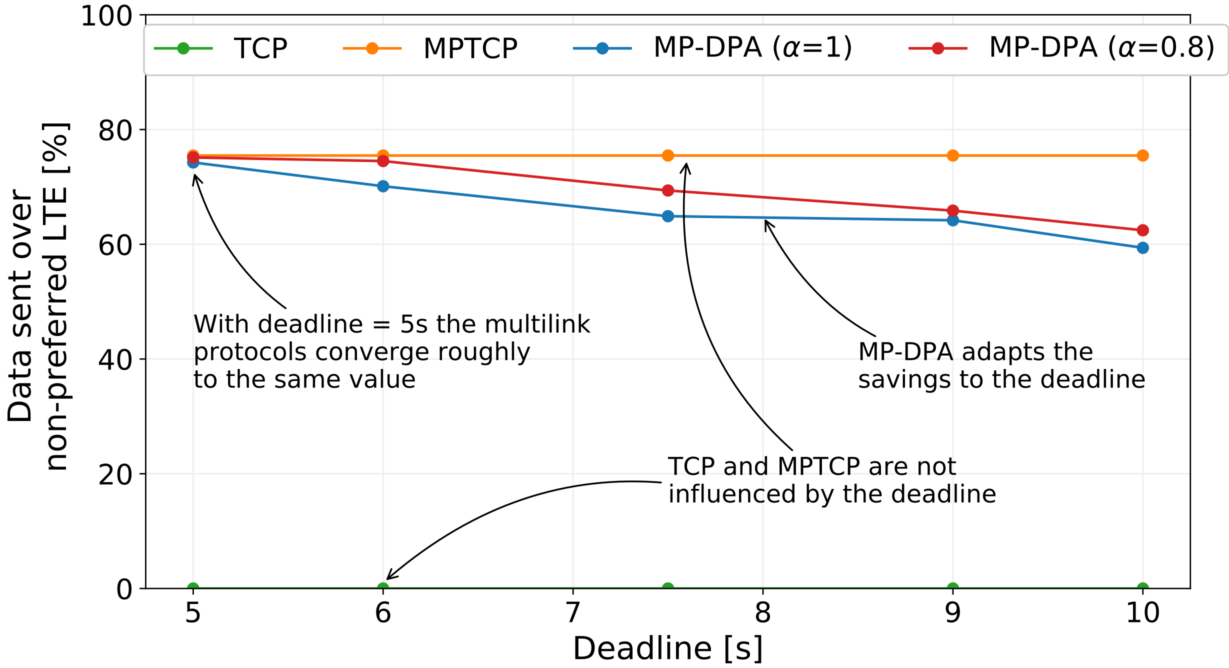


(a) Except for TCP that never uses the second interface, the other frameworks exploits it differently. In MPTCP the data is equally distributed independently from the deadline used. MP-DPA savings are directly proportional to the value of the deadline and α. All the multipath protocols converge to $\sim$50% for a deadline of 5 seconds.

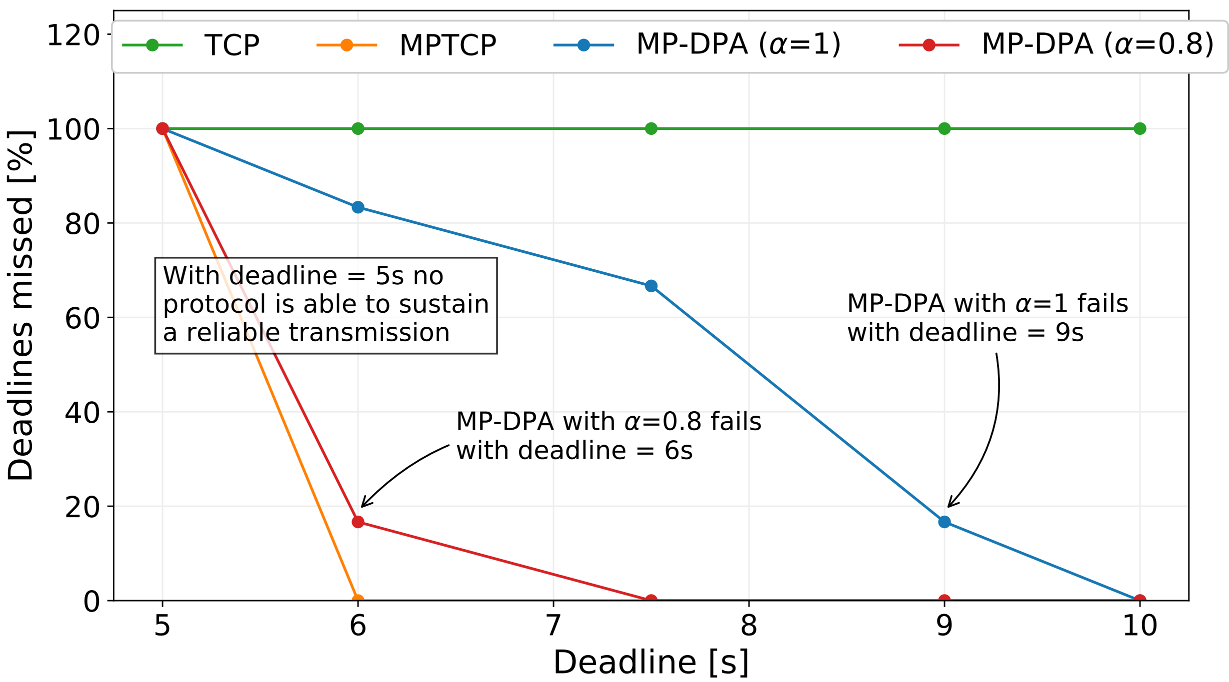


(b) MP-DPA with $\alpha = 1$ shows its limit at 6 seconds of deadline. By decreasing its value to 0.8 the reliability is the same of MPTCP. It is ensured till a deadline of 5 seconds where the upload of every packet fails.

Figure 6.15: Definition of the limits for TCP, MPTCP, MP-DPA (α=1) and MP-DPA (α=0.8) in symmetric conditions. **(a)** Percentages of data sent over the non-preferred interface. **(b)** Percentages of deadline missed.



(a) Each protocol shows globally higher percentages of data sent over non-preferred interfaces. The savings, although present, are no more considerable like the symmetric case.



(b) MP-DPA with α starts to fail the transmission for a deadline of 7.5 seconds. Even if α is reduced to 0.8, a packet is not fully uploaded for 6 seconds of deadline. To obtain the same results of MPTCP, must be considered an even lower value of α.

Figure 6.16: Definition of the limits for TCP, MPTCP, MP-DPA (α=1) and MP-DPA (α=0.8) in asymmetric conditions. **(a)** Percentages of data sent over the non-preferred interface. **(b)** Percentages of deadline missed.

6.5 Investigation on the Deadline Sensitivity

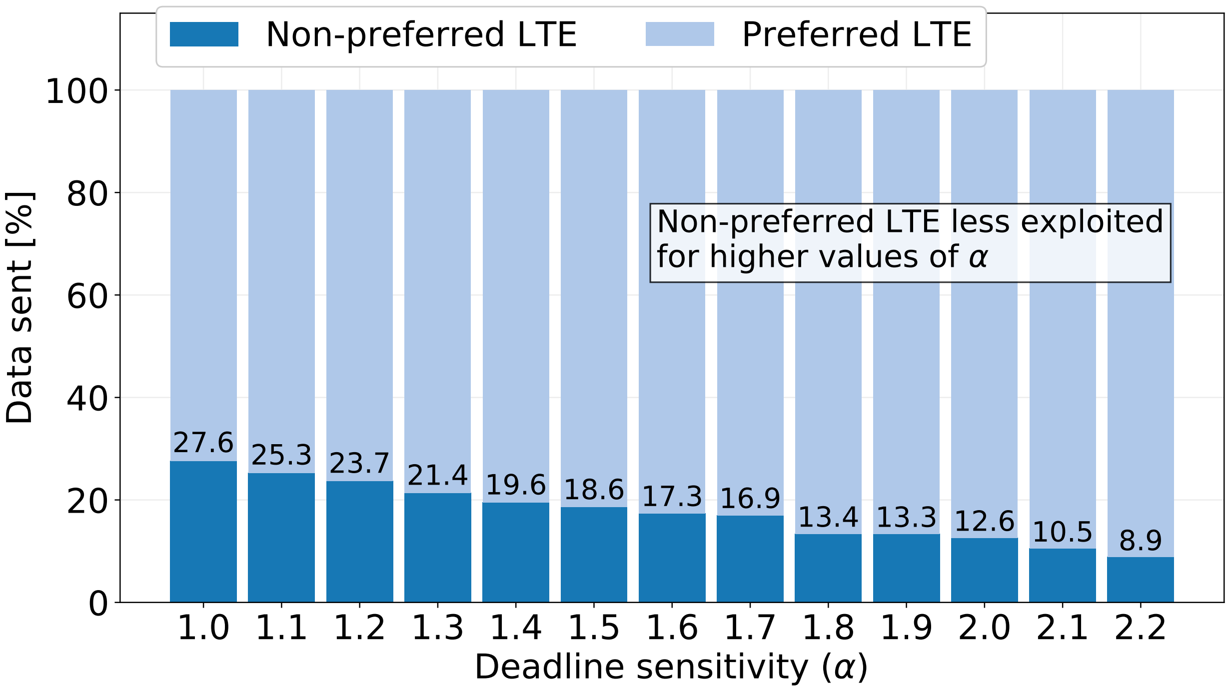
In this section MP-DPA is in-depth analyzed to find the best α that fits for a given network configuration in static conditions. It is achieved by defining the utility factor μ. It quantified the effectiveness of MP-DPA in a given condition.

6.5.1 Main Metrics Analysis

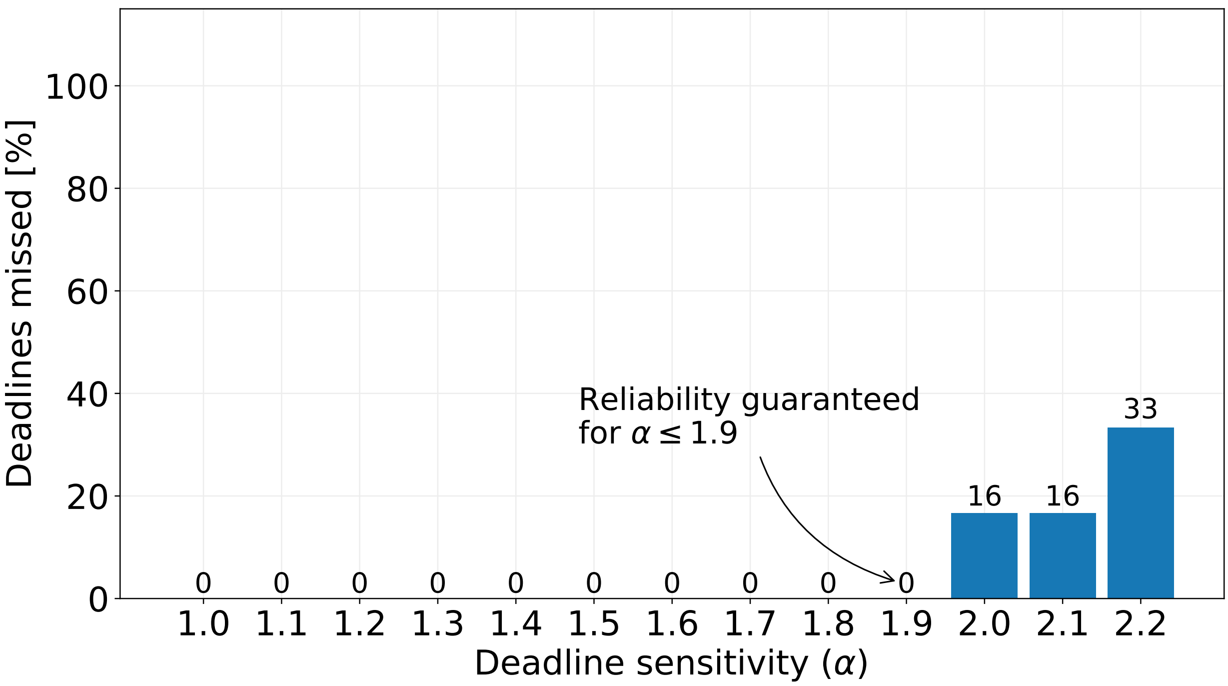
Since MP-DPA is designed to minimize data over non-preferred interface while respecting every deadline, in the computation of μ, both properties must be considered.
The following set of images refer to the analysis of MP-DPA in terms of non-preferred LTE usage and deadline missed for symmetric and asymmetric bandwidths, respectively. In particular, for every combination of network type and deadline value, are compared the results for multiple values of α. As before, for all the experiments, are transmitted 60 MB of data divided into 6 packets of 10 MB each, everyone sent every 10 seconds.

In the case of symmetric bandwidths, since MP-DPA is in favorable conditions, a higher value of α (> 1) is more preferable that a lower one. From the data savings in both top graphs of Figure 6.17 and Figure 6.18, despite the heterogeneity of the scenarios, can be identified a recurring trend. By increasing α, the virtual deadline seen by the sender increases, and thus, less frequently the non-preferred interface is enabled. Consequently, with higher values of α, the savings are more considerable but more deadlines are not respected. In fact, in case of a deadline = 10 seconds (Figure 6.17 **(b)**), all the packets are entirely sent till $\alpha = 1.9$. For 7.5 seconds (Figure 6.18 **(b)**) instead, only a lower value of 1.1 does not compromise the reliability.

When MP-DPA is forced to use more the non-preferred interface in asymmetric conditions, a lower value of α is preferred. In terms of data savings (**(a)** images in Figure 6.19 and Figure 6.20), the resulting behavior is the same observed in symmetric conditions. With a bigger deadline sensitivity, the non-preferred LTE is used less.
In both figures **(b)** instead, the deadlines missed are already considerable for $\alpha = 1.1$, the best value for symmetric conditions. With deadline = 10 seconds in Figure 6.19 **(b)** and 7.5 seconds in Figure 6.19 **(b)**, the deadlines are satisfied only for $\alpha < 1$: 0.9 and 0.8, respectively.
To conclude, the more MP-DPA is in stressed conditions (from symmetric with a soft deadline to asymmetric with a strict deadline), the smaller is the interval of α that satisfied all the deadlines. It means that for bad network conditions, the value of α is of particular importance.

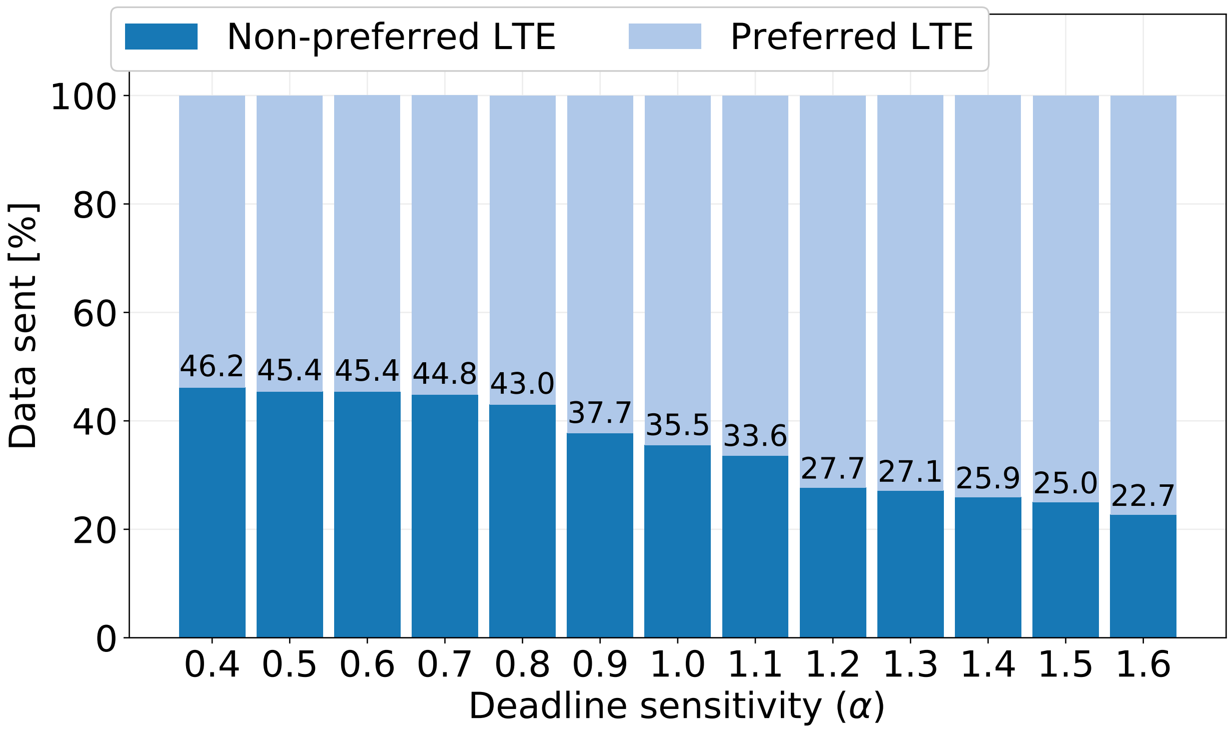


(a) With higher values of α, the non-preferred interface is activated less frequently.

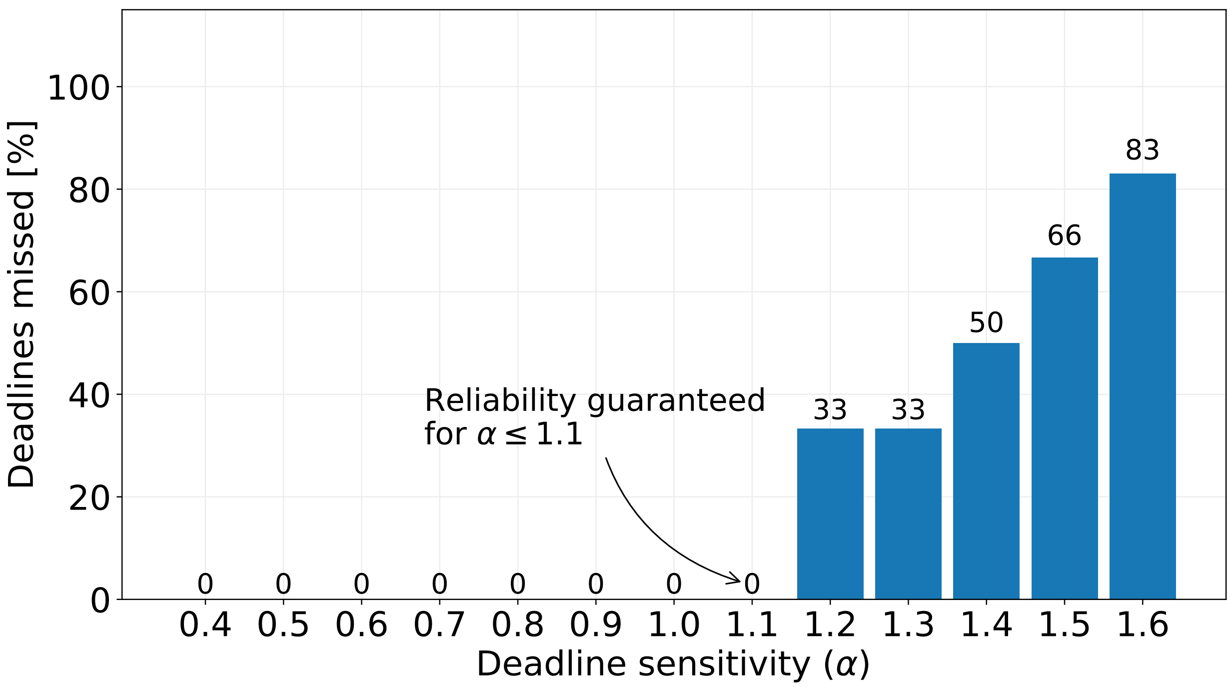


(b) The deadlines are missed more likely for higher values of α. The possible best α values is 1.9 for a relaxed deadline.

Figure 6.17: Comparison of MP-DPA data distribution over the LTEs **(a)** and the deadlines missed in symmetric conditions for multiple α values with deadline = 10 s **(b)**.

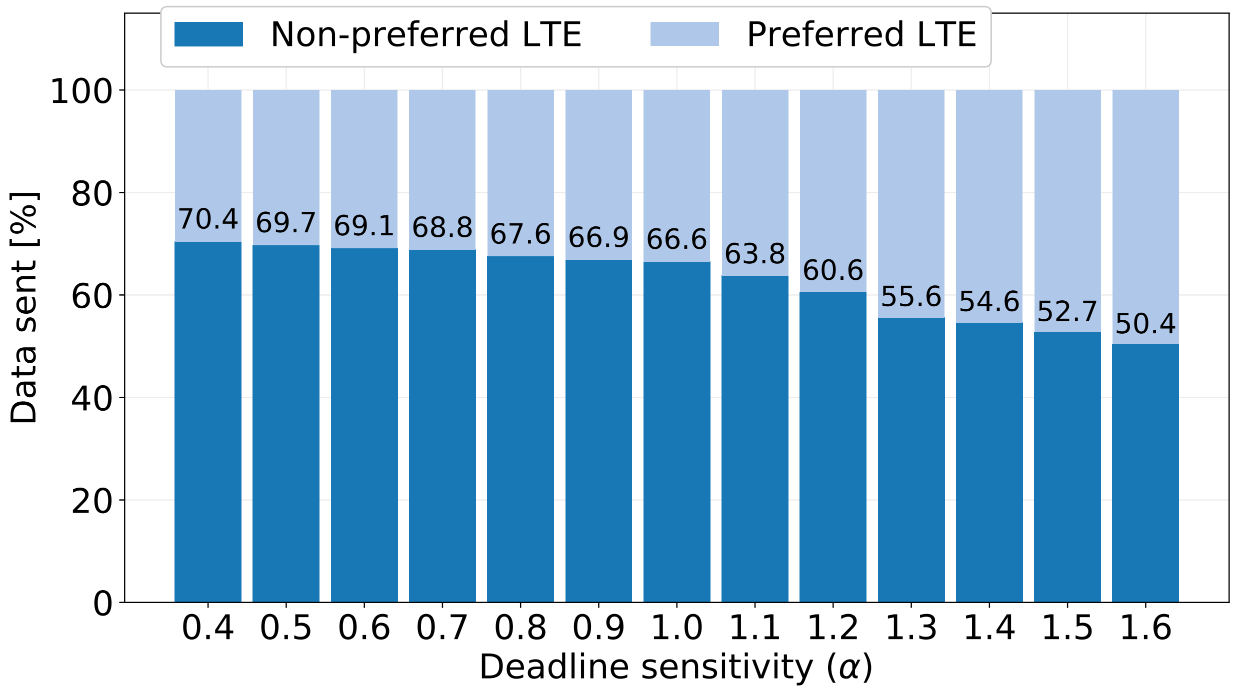


(a) With higher values of α, the non-preferred interface is activated less frequently.

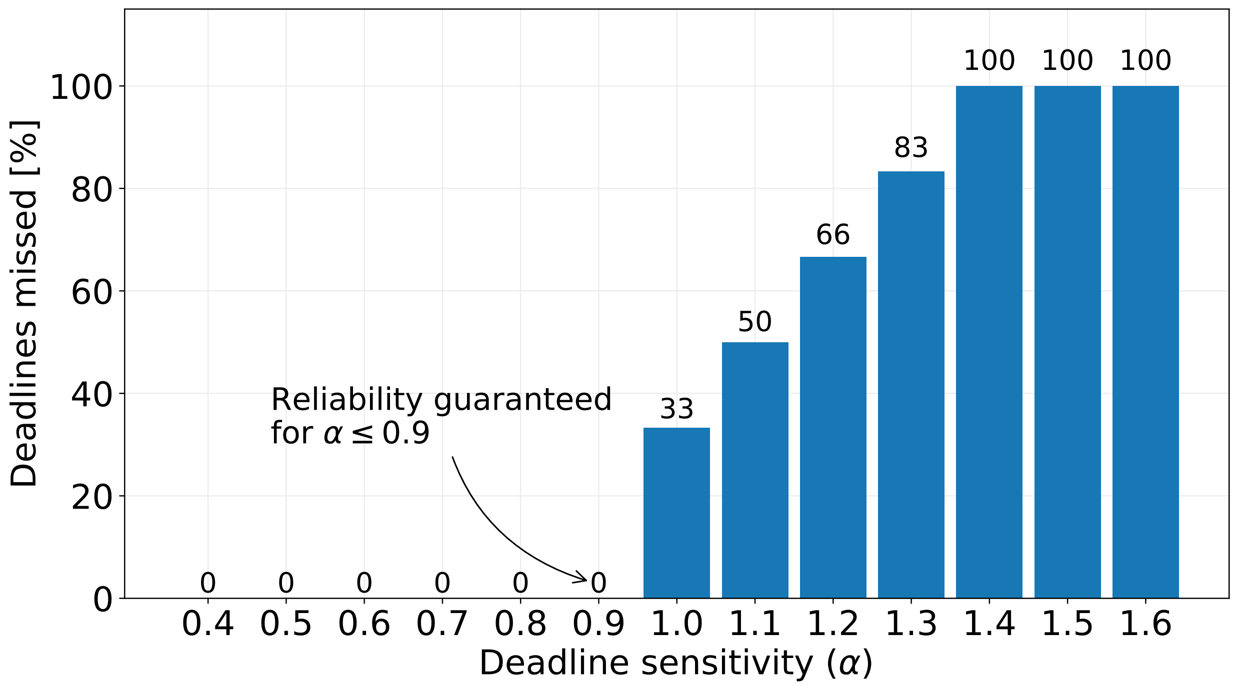


(b) The deadlines are missed more likely for higher values of α. The possible best α values is 1.1 for a strict deadline.

Figure 6.18: Comparison of MP-DPA data distribution over the LTEs **(a)** and the deadlines missed in symmetric conditions for multiple α values with deadline = 7.5 s **(b)**.

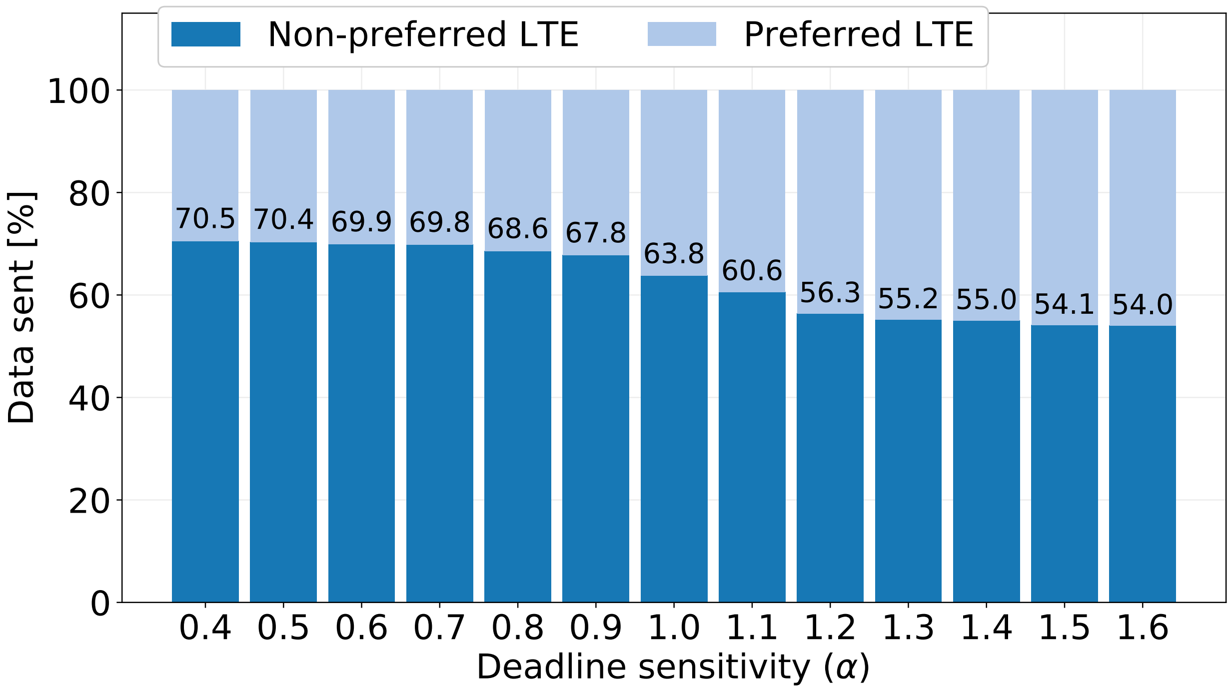


(a) Globally, more data are sent over the second LTE. However, with higher values of α, the non-preferred interface is activated less frequently.

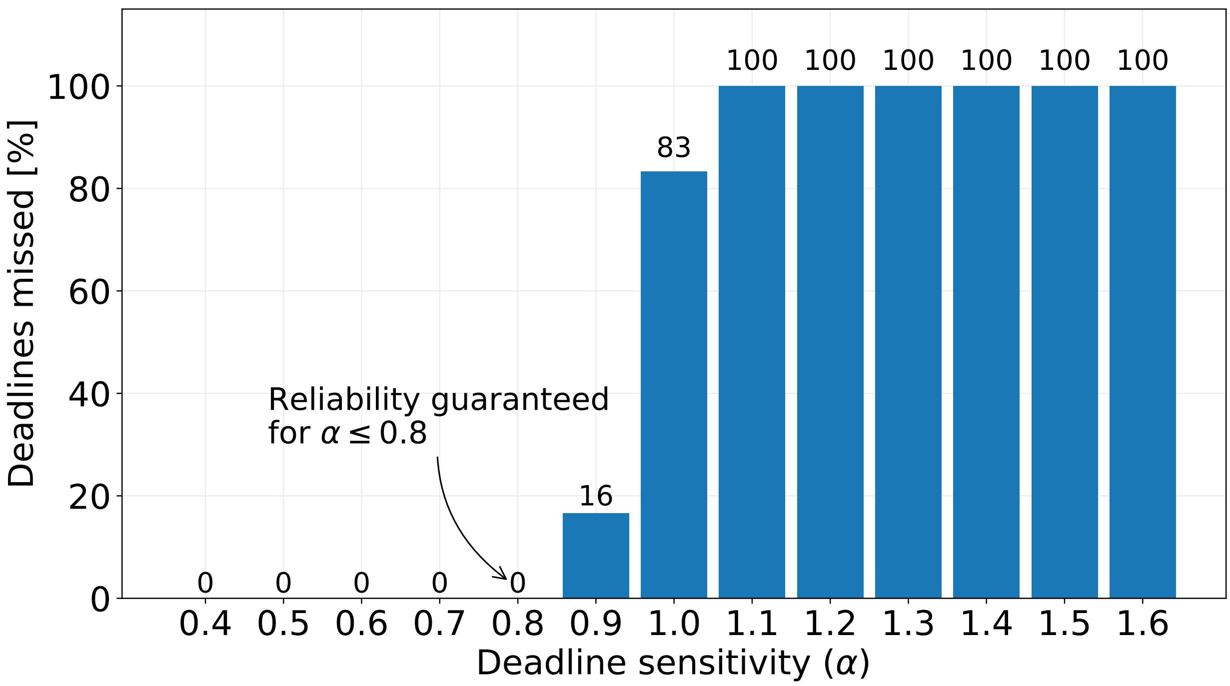


(b) The range of α that fulfills the reliability is restricted for lower values. The deadlines are missed more likely for higher values of α. The possible best α values is 0.9 for a relaxed deadline.

Figure 6.19: Comparison of MP-DPA data distribution over the LTEs **(a)** and the deadlines missed in asymmetric conditions for multiple α values with deadline = 10 s **(b)**.



(a) Globally, more data are sent over the second LTE. However, with higher values of α, the non-preferred interface is activated less frequently.



(b) The range of α that fulfills the reliability is restricted for lower values. The deadlines are missed more likely for higher values of α. The possible best α values is 0.8 for a strict deadline.

Figure 6.20: Comparison of MP-DPA data distribution over the LTEs **(a)** and the deadlines missed in asymmetric conditions for multiple α values with deadline $= 7.5$ s **(b)**.

6.5.2 Data Overhead Estimation

To analyze the effectiveness of MP-DPA in its entirety, any possible aspect that may influence negatively the performances must be considered. For this purpose, the main goals of MP-DPA are put aside and the focus is shifted to the total amount of MB sent to the receiver. Firstly, are identified the differences between a standard TCP transmission while successively, MP-DPA overhead is analyzed for multiple scenarios.

TCP and MultiPath Protocols Differences

Figure 6.21 compares how many MB are completely sent for each protocol with multiple deadlines for both symmetric **(a)** and asymmetric **(b)** bandwidths. To quantify the overhead, it is demonstrated, by multiple TCP simulations with 60 MB of payload, that TCP sends an overhead of data equal to 2% of the total (∼1.2 MB). This value is represented by the red line.

Except for TCP which does not respect deadlines and thus, stops the transmission of the packets, MPTCP and MP-DPA exhibit an overhead of data sent. If for MPTCP this surplus of data is regular, MP-DPA overhead grows with the increasing of the deadlines values.

The higher MP-DPA overhead is caused by the enabling and disabling procedure of the non-preferred interface. As can be noticed in Figure 6.3 and Figure 6.11, MP-DPA throughput is more variable than MPTCP one. At the beginning and end of every packet (for MPTCP) and also every time the second LTE is switched on and off (for MP-DPA), is transmitted a small overhead of data that does not belong to the effective payload. At the end of the simulation, after 60 MB of payload sent, the data glut exhibited by MPTCP is 6.6% while MP-DPA one is between 10% and 14%.

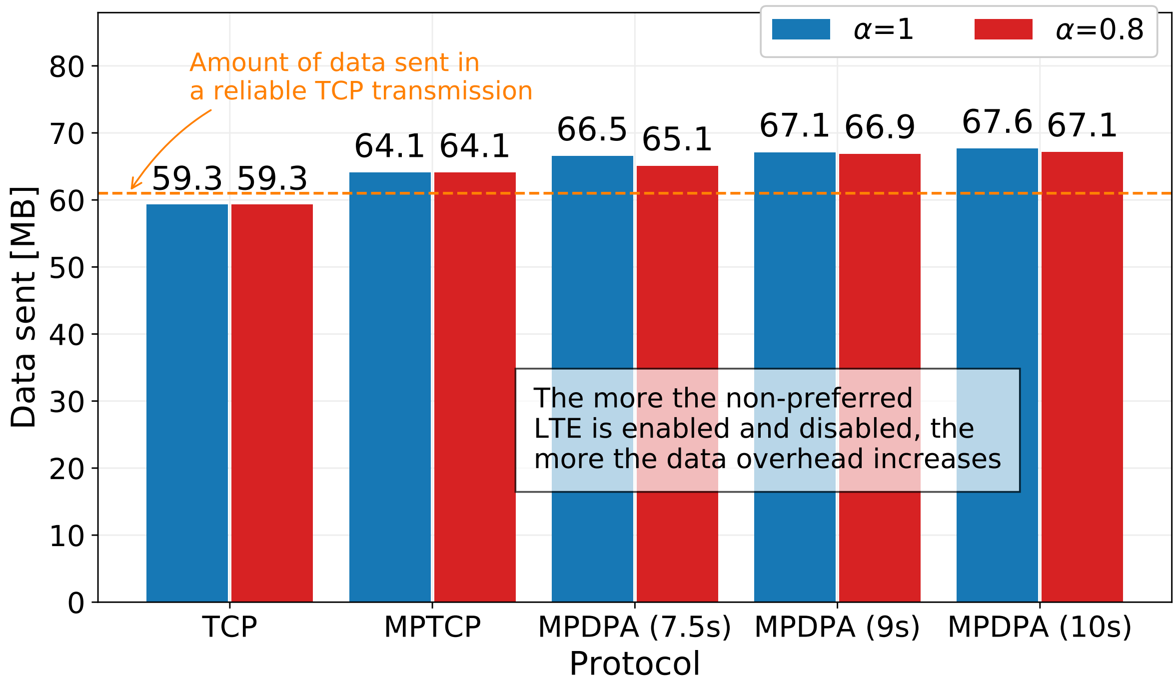
Observing the red bars of Figure 6.21, with a lower α value, the non-preferred LTE is enabled for more time, and thus the enabling or disabling procedure is less frequent. For this reason, as demonstrated by both symmetric and asymmetric results, by reducing α, the amount of data sent as overhead is slightly decreased.

MP-DPA Data Overhead

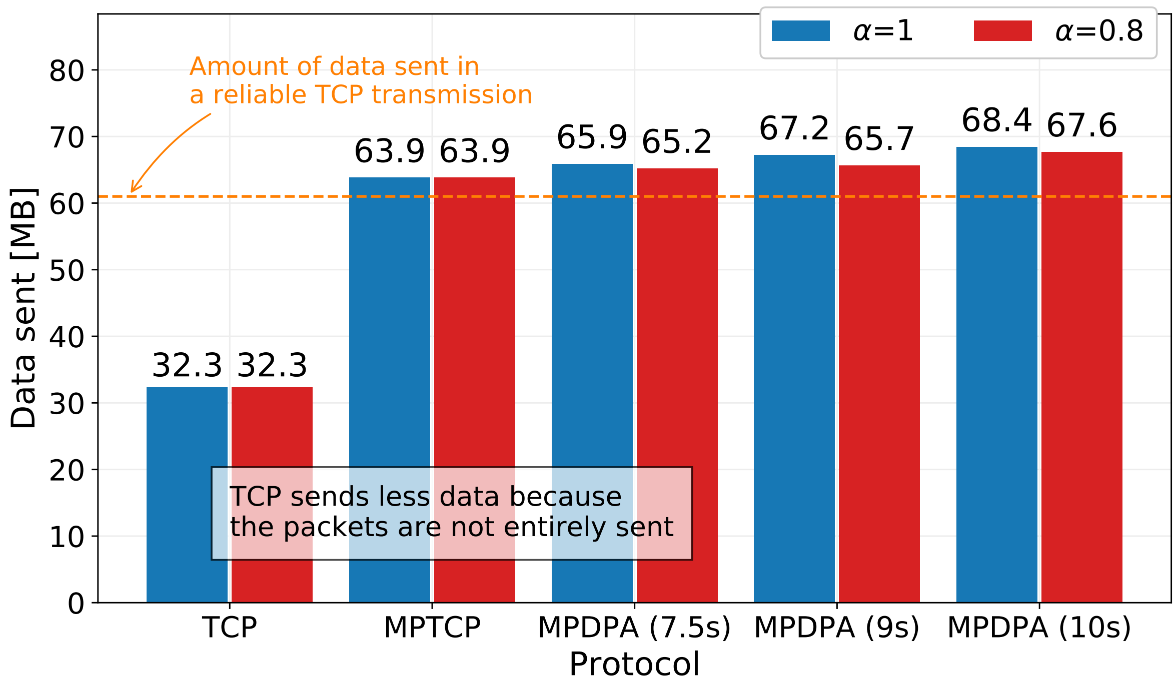
Since the utility factor takes into account the effectiveness of MP-DPA, we need to analyze also the difference of data overhead between MPTCP and MP-DPA. Figure 6.22 and Figure 6.23, respectively for symmetric and asymmetric bandwidths, express in percentage the difference of transmitted data between MP-DPA and MPTCP with respect to the total amount of data sent through MPTCP.

The results obtained in every experiment exhibit a Gaussian dependence between α and the data overhead. In the case of Figure 6.22 **(a)** with symmetric bandwidths and deadline = 10 seconds, MP-DPA overhead increases with a lower values of α. Anyway, the plot shows only the right-last part of the Gaussian waveform, since these optimal conditions allow MP-DPA to achieve the best performances with very high values of α.

In all the other experiments, where the observed α interval is lower, the complete Gaussian shape is more accentuated. Approximately in the middle of Figure 6.22 **(a)** and Figure 6.23 can be identified an overhead peak. The lower is the value of α and longer the non-preferred LTE stays on, resulting

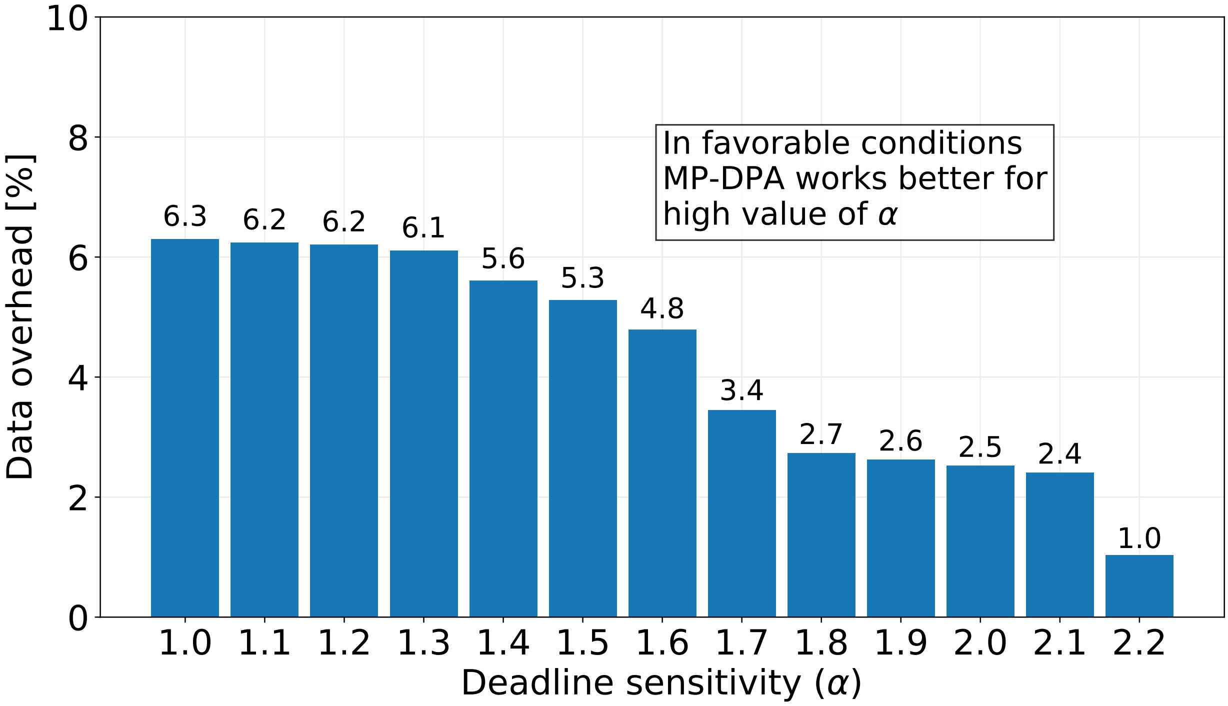


(a) Symmetric conditions.

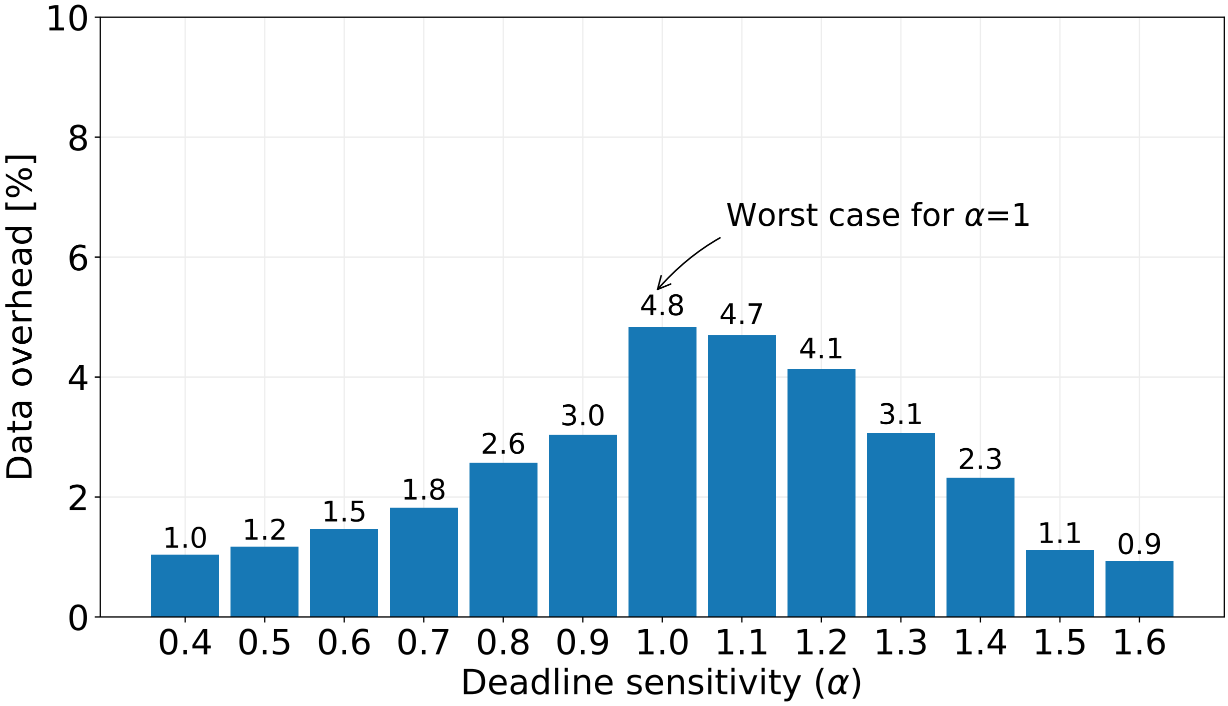


(b) Asymmetric conditions.

Figure 6.21: Comparing the total amount of data sent in TCP, MPTCP and MP-DPA with different deadlines with symmetric **(a)** and asymmetric **(b)** bandwidths. The orange line indicates the effective data sent during a reliable TCP transmission. The more the enabling and disabling procedure of the non-preferred interface is frequent, the more it leads to an overhead. TCP sends less data because when the deadline of a packet is missed, its transmission is interrupted.

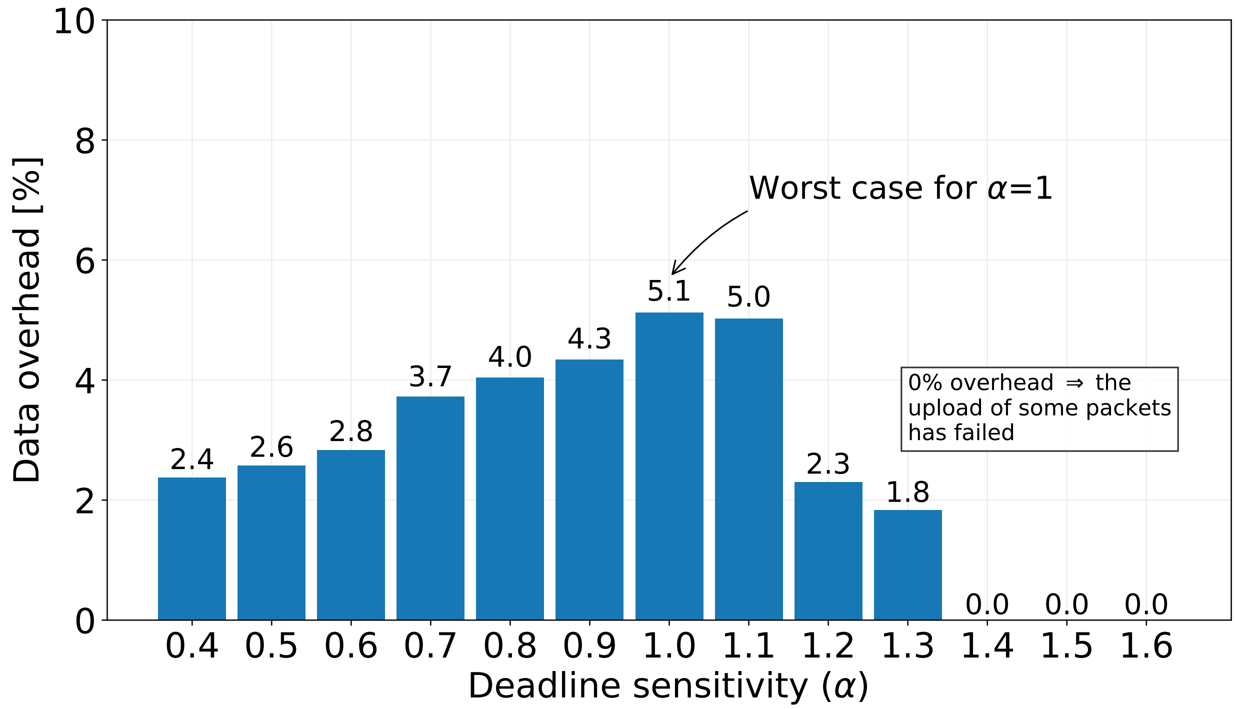


(a) Symmetric bandwidths and deadline=10 seconds.

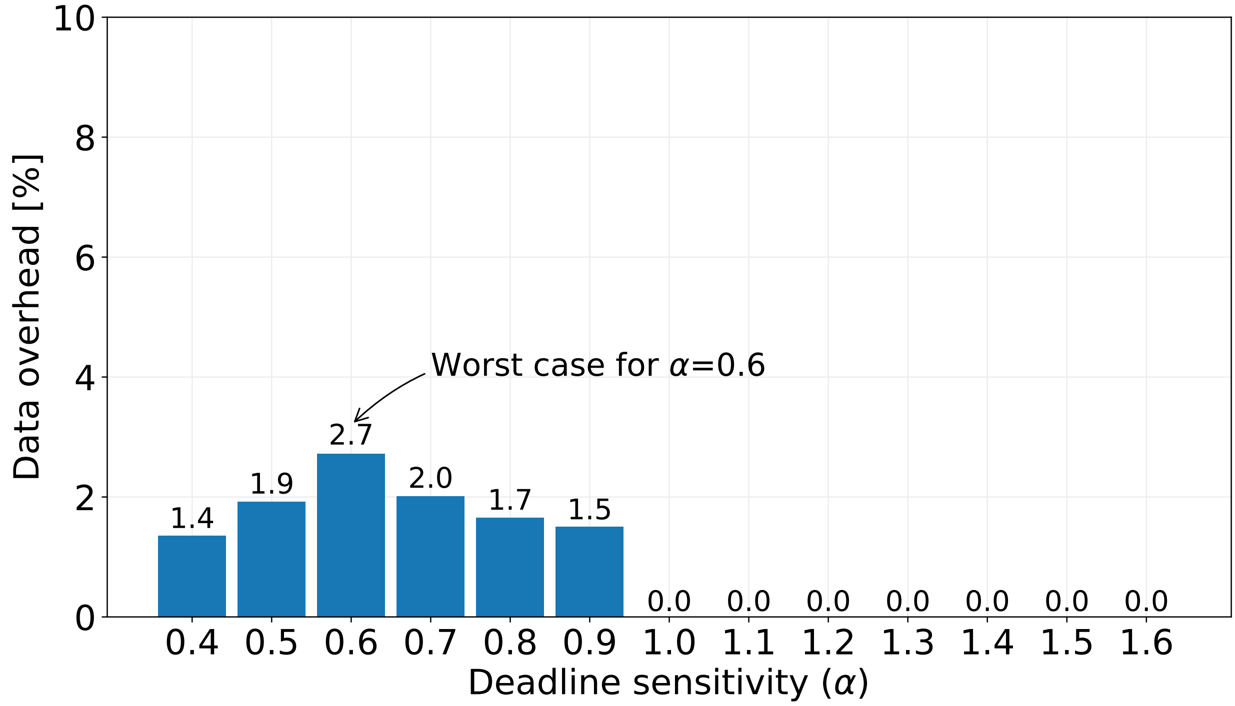


(b) Symmetric bandwidths and deadline=7.5 seconds.

Figure 6.22: Measurements of data overhead of MP-DPA compared to MPTCP for multiple α values in symmetric conditions. The graph in **(b)** exhibits a Gaussian-like dependence, while **(a)** displays only its right side. The more the enabling and disabling procedure of the non-preferred LTE is frequent, the bigger is the overhead.



(a) Asymmetric bandwidths and deadline=10 seconds.



(b) Asymmetric bandwidths and deadline=7.5 seconds.

Figure 6.23: Measurements of data overhead of MP-DPA compared to MPTCP for multiple α values in asymmetric conditions. The graphs exhibit a Gaussian-like dependence. The more the enabling and disabling procedure of the non-preferred LTE is frequent, the bigger is the overhead. The 0 value represents a non-reliable transmission.

in a less frequent enabling and disabling procedure. Moving toward the middle of the Gaussian function, with higher α, the second LTE is switched on and off more often, wit a consequent bigger overhead. In the last part of the graphs instead, the overhead decreases because the non-preferred interface is exploited less and less till reaching a value where it is even not exploited (TCP-like transmission).

In case MP-DPA does not respect the deadlines, for example, due to a too high α value, the transmission of the packet is interrupted. It results in a lower amount of data sent than MPTCP. Instead of representing negative values of overhead, this behavior is represented by the 0 values.

It is important to notice that the more MP-DPA has to work in not optimal conditions, the more the Gaussian waveform is shifted to lower α values.

6.5.3 Utility Factors Computation

As detailed in 5.1.3, the main goal of μ is to investigate the best α value that maximizes the effectiveness of MP-DPA. Here are considered two utility factors: one is more focused on the reliability aspect of MP-DPA while the second one is more centered on the non-preferred interface usage minimization.

In the best case, when μ=1, no deadlines are missed, no data are sent over the second interface and no overhead is present compared to MPTCP.

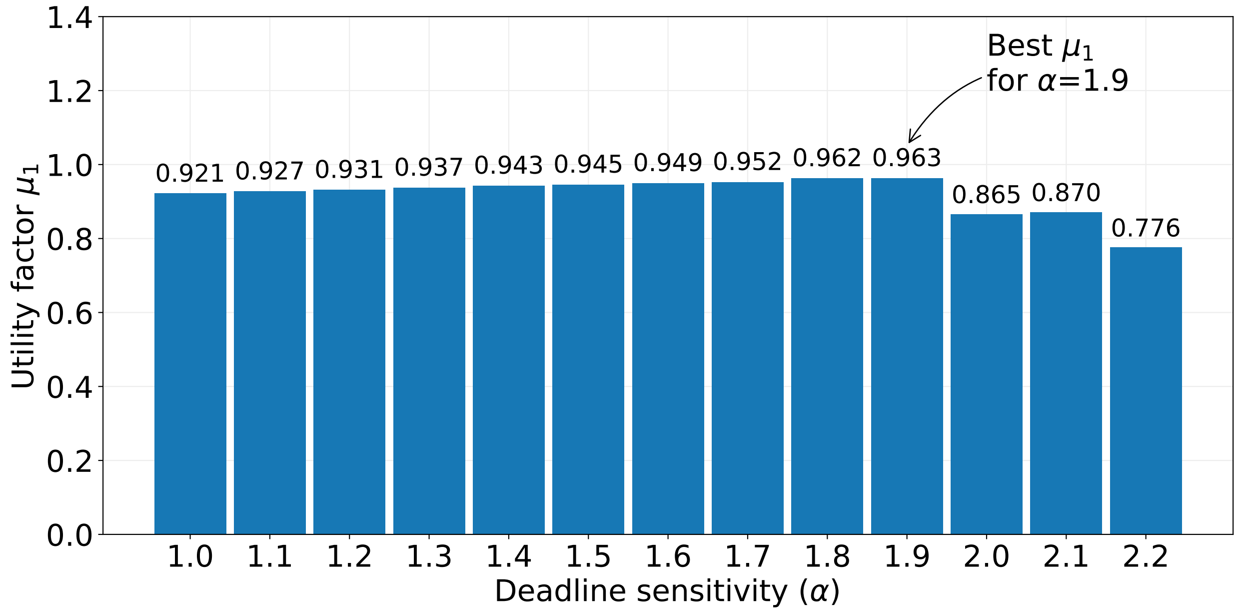
Utility Factor μ_1 for Reliability Purposes

For μ_1 computation, MP-DPA efficacy is evaluated considering the percentage of deadlines missed as metric of major interest. By the combination of the previous results, depending on the network configuration, α assumes for every experiment different pareto-optimal values as depicted in Figure 6.24 and Figure 6.25 for symmetric and asymmetric conditions, respectively.

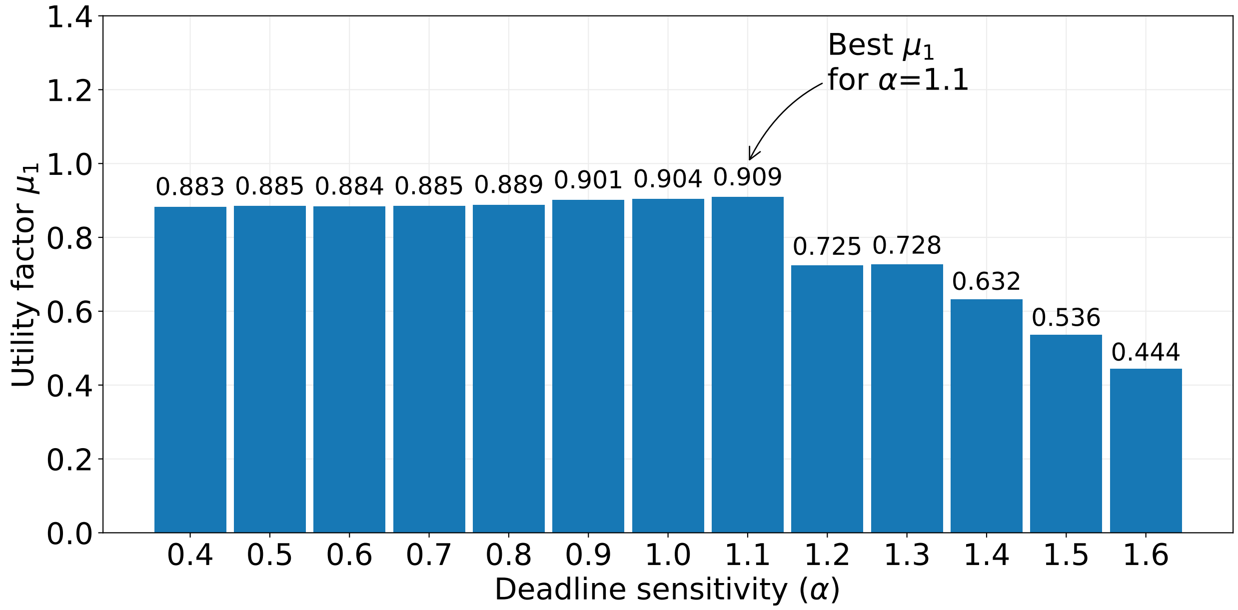
From a global vision of all the plots, it is easy to identify the best value of α since it is always located before the one for which one or more deadlines are not respected. In case the reliability is compromised, μ decreases very rapidly to low values.

For symmetric bandwidths in Figure 6.24 with 10 **(a)** and 7.5 seconds of deadline **(b)**, α must be set to 1.9 and 1.1, respectively, in order to achieve the MP-DPA best performances. In the asymmetric case in Figure 6.25 instead, μ_1 shows its peak when the deadline sensitivity assumes lower values such as 0.9 when the deadline is 10 seconds **(a)** and 0.8 with a deadline of 7.5 seconds **(b)**.

The more MP-DPA is used in critical conditions such as asymmetric ones with a strict deadline, the lower must be configured α to have an optimal MP-DPA behavior. Thus, if the main objective is to maximize the reliability of the protocol in static conditions with bulk upload, the deadline sensitivity α must be set to 0.8 to fully exploit MP-DPA advantages.

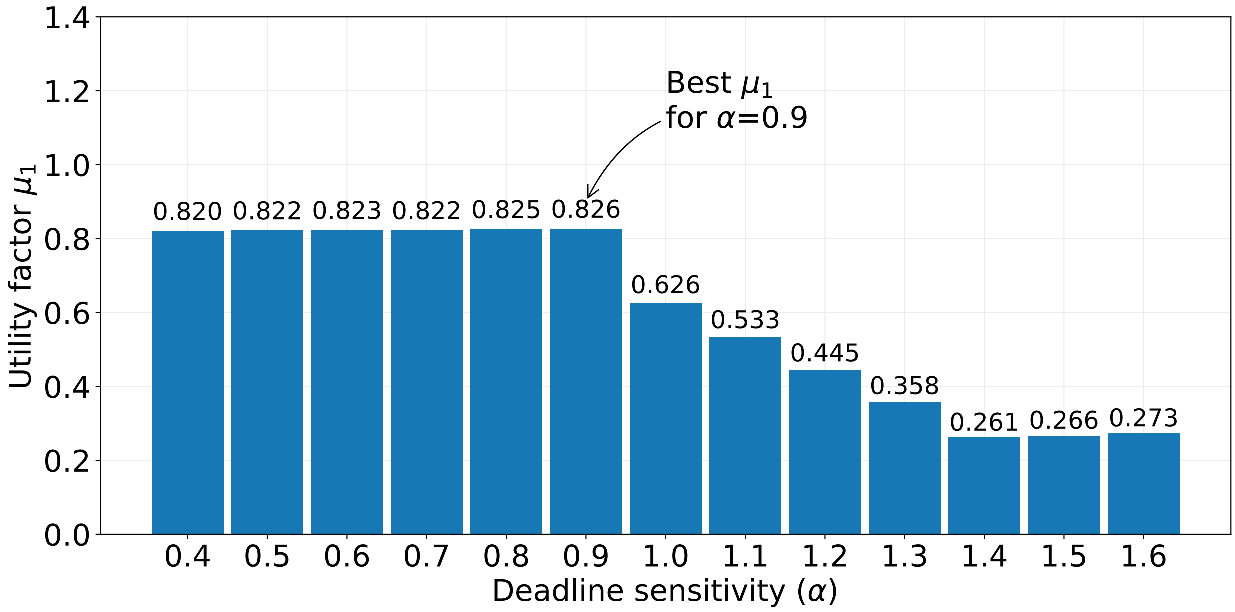


(a) Symmetric bandwidths and deadline=10 seconds.

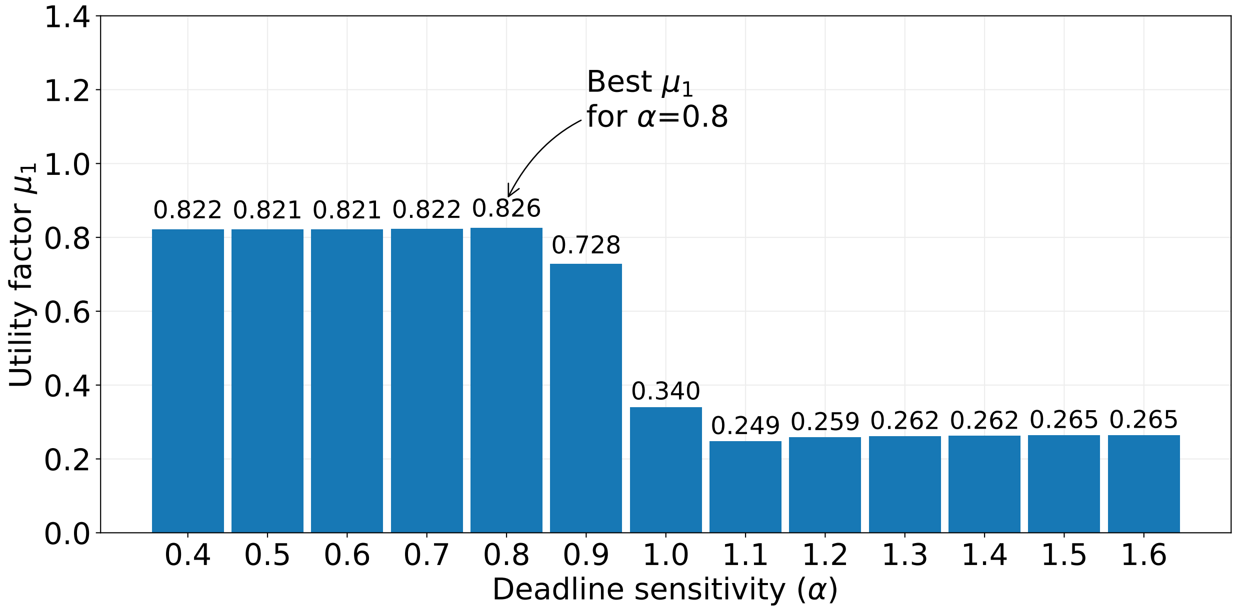


(b) Symmetric bandwidths and deadline=7.5 seconds.

Figure 6.24: Comparing for multiple α values the utility factor μ_1 in symmetric conditions. In case one or more deadlines are not respected, the effectiveness of MP-DPA decreases considerably. The best α value is always located before the one for which one or more deadlines are not respected. The best values for α are 1.9 and 1.1 for 10 and 7.5 seconds as deadline, respectively.



(a) Asymmetric bandwidths and deadline=10 seconds.

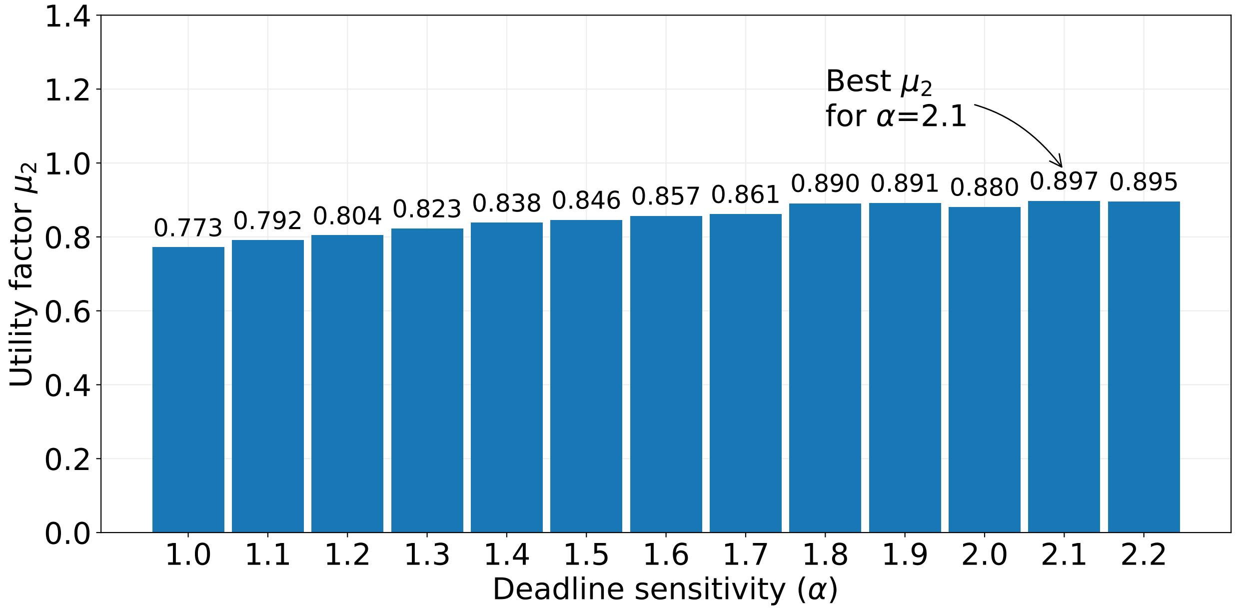


(b) Asymmetric bandwidths and deadline=7.5 seconds.

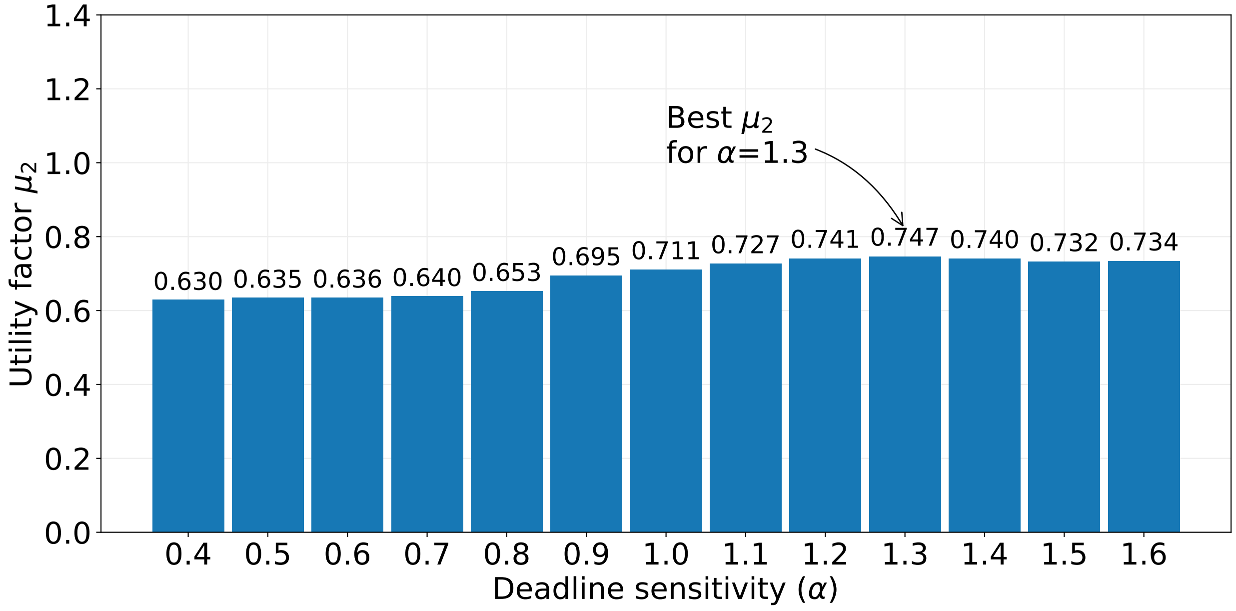
Figure 6.25: Comparing for multiple α values the utility factor μ_1 in asymmetric conditions. In case one or more deadlines are not respected, the effectiveness of MP-DPA decreases considerably. The best α value is always located before the one for which a deadline is not respected. The best values for α are 0.9 and 0.8 for 10 and 7.5 seconds as deadline, respectively.

Utility Factor μ_2 for Non-Preferred Interface Usage Minimization

Similar observations can be done for the second utility factor μ_2 where the minimization of non-preferred LTE usage is considered as more relevant metric. The results are shown in Figure 6.26



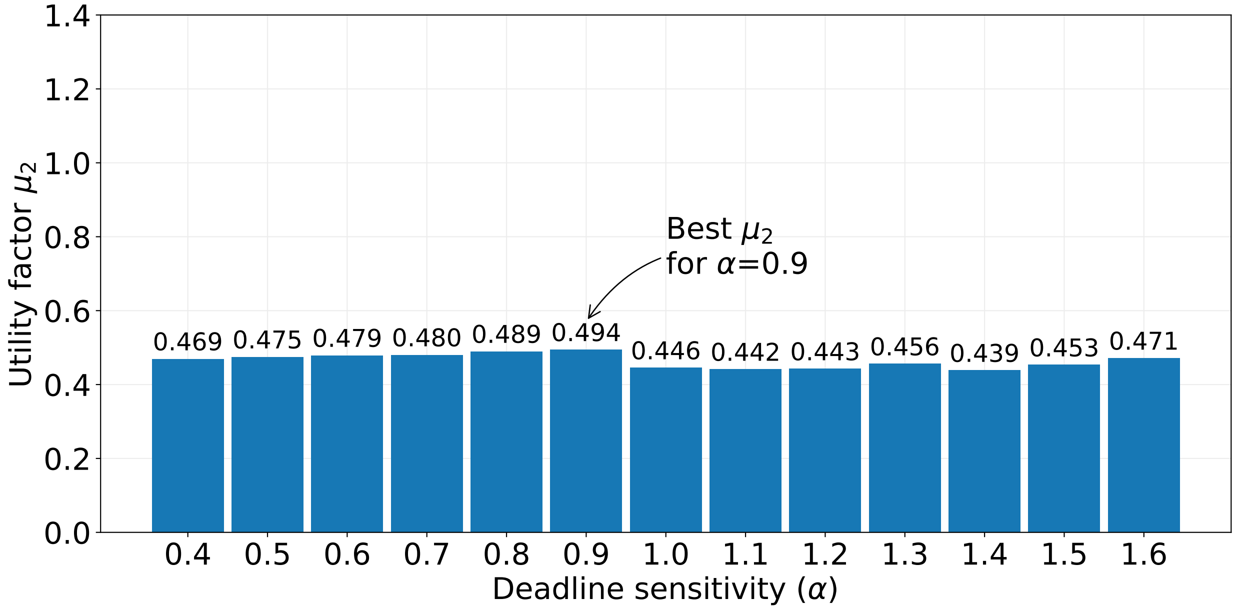
(a) Symmetric bandwidths and deadline=10 seconds.



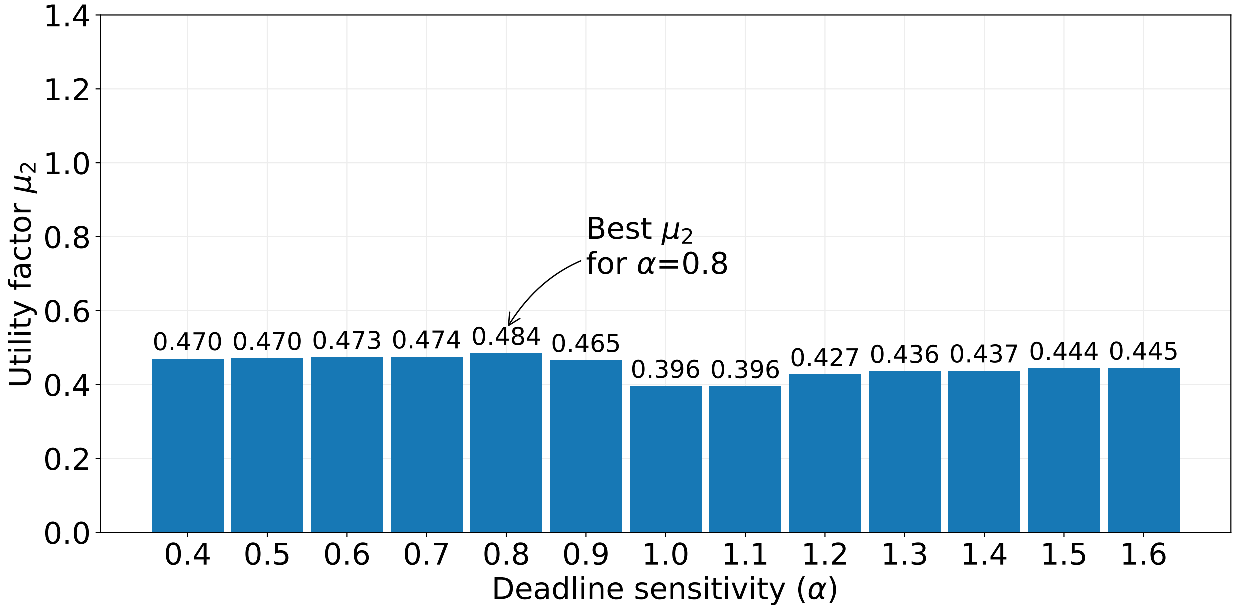
(b) Symmetric bandwidths and deadline=7.5 seconds.

Figure 6.26: Comparing for multiple α values the utility factor μ_2 for symmetric bandwidths. The results have a more uniform shape, thus the best α value is hardly recognisable. For high values of α, despite all the deadlines are missed, μ_2 exhibits similar values to the lower ones since they are characterized by large savings. The pareto-optimal α values are bigger than the ones computed in Figure 6.24.

and Figure 6.27 for symmetric and asymmetric bandwidths. Globally, μ_2 values are more evenly distributed compared to μ_1 but the best α can still be identified.



(a) Asymmetric bandwidths and deadline=10 seconds.



(b) Asymmetric bandwidths and deadline=7.5 seconds.

Figure 6.27: Comparing for multiple α values the utility factor μ_2 for asymmetric bandwidths. The results have a more uniform shape, thus the best α value is hardly recognisable. For high values of α, despite all the deadlines are missed, μ_2 exhibits similar values to the lower ones since they are characterized by large savings. The pareto-optimal α values coincide to the ones computed in Figure 6.25.

For high values of α, despite all the deadlines are missed, μ_2 exhibits similar values to the lower ones since they are characterized by large savings. Although the reliability aspect is not considered as the main metric, a high number of deadlines missed are still able to decrease the value of μ_2.

This explains why the optimal value of α is not located at the far right of the graph, but it is still near to the one computed for μ_1.
Using symmetric bandwidths (Figure 6.26), the pareto-optimal values of α are shifted of two decimal positions to the right. For 10 and 7.5 seconds of deadlines, α must be set to 2.1 and 1.3, respectively. If the network is arranged to be asymmetric (Figure 6.27), μ_2 peaks are located at the same α values of μ_1.
From the deadline sensitivity analysis, can be concluded that does not exist an optimal value of α that works for every network condition. However, if any information such as bandwidth and deadlines are known, α can be modified accordingly to them. In this way, the deadline sensitivity can be adapted to the current scenario, and thus, MP-DPA can be more effective. In real cases, since the network conditions are not known but depend on the current vehicle location, for the next experiments α is set to 0.8, the best value in the worst-case scenario.

6.6 From Bulk to Streaming Transmission in Pseudo-dynamic Conditions

As proved with the deadline sensitivity analysis, MP-DPA can reduce the overall cost of the transmission while maintaining the same good performance of a multipath protocol such as MPTCP. By modifying accordingly the value of the deadline sensitivity α, MP-DPA can be adapted to the current network conditions. However, according to the conclusions about μ in 6.5.3, for the next experiments the deadline sensitivity α is fixed to 0.8.
Till now, the Multipath Deadline and Preference Aware (MP-DPA) is in-depth evaluated and validated in a bulk transmission typical of a car-to-cloud communication. To analyze more deeply the advantages of the new framework, the sections below report the results using a streaming transmission. During a streaming upload, the idle time during consecutive packets, proper of bulk transmissions, is not present, but the vehicle sends the data as soon as they are acquired. From a simulation point of view, this behavior is replicated by reducing considerably the time between the transmission of two consecutive packets.

6.6.1 Basic Performances Assessment

Being the bandwidth of the preferred LTE configured with a triangular shape, it is expected that MP-DPA enables the non-preferred one close to the minimum values. Figure 6.28 illustrates the effective throughput for all the protocols with a deadline of 1 second.
In every graph, the streaming upload is characterized by a much more variable throughput than the bulk upload results in Figure 6.3. For every packet, MPTCP tries to fully exploit both interfaces to reach the maximum speed. However, can be observed the presence of spikes that do not reach the full available bandwidth since before it is reached, a new packet is scheduled. Thus, at the end of the transmission of a packet, the interfaces return to the idle state. Besides, since a high number of packets are sent in a relatively little amount of time, the enable and disable procedure of both interfaces occurs more frequently causing unstable behavior.
As predicted, MP-DPA uses the non-preferred interface in correspondence of the local minimums of the triangular waveform, where the preferred LTE lacks is performances. In particular, for a deadline of 1 second, the second LTE helps the transmission of the packets at the beginning and in the middle of the communication.
Despite the effective throughput is slightly different from the bulk scenario, the overall behavior considering the average time needed to send a packet is the same. It is displayed in Figure 6.29.
For standard TCP transmissions, as the only first interface is available, the communication is slower than multilink protocols. Since MPTCP utilizes both interfaces, its latency is instead very low ($\sim$0.06 seconds) compared to the sending frequency of 1 second. For both protocols, although the deadline changes, the time needed for a packet is quite stable.
This consideration cannot be done for MP-DPA. The adaptive behavior of MP-DPA can be indi-

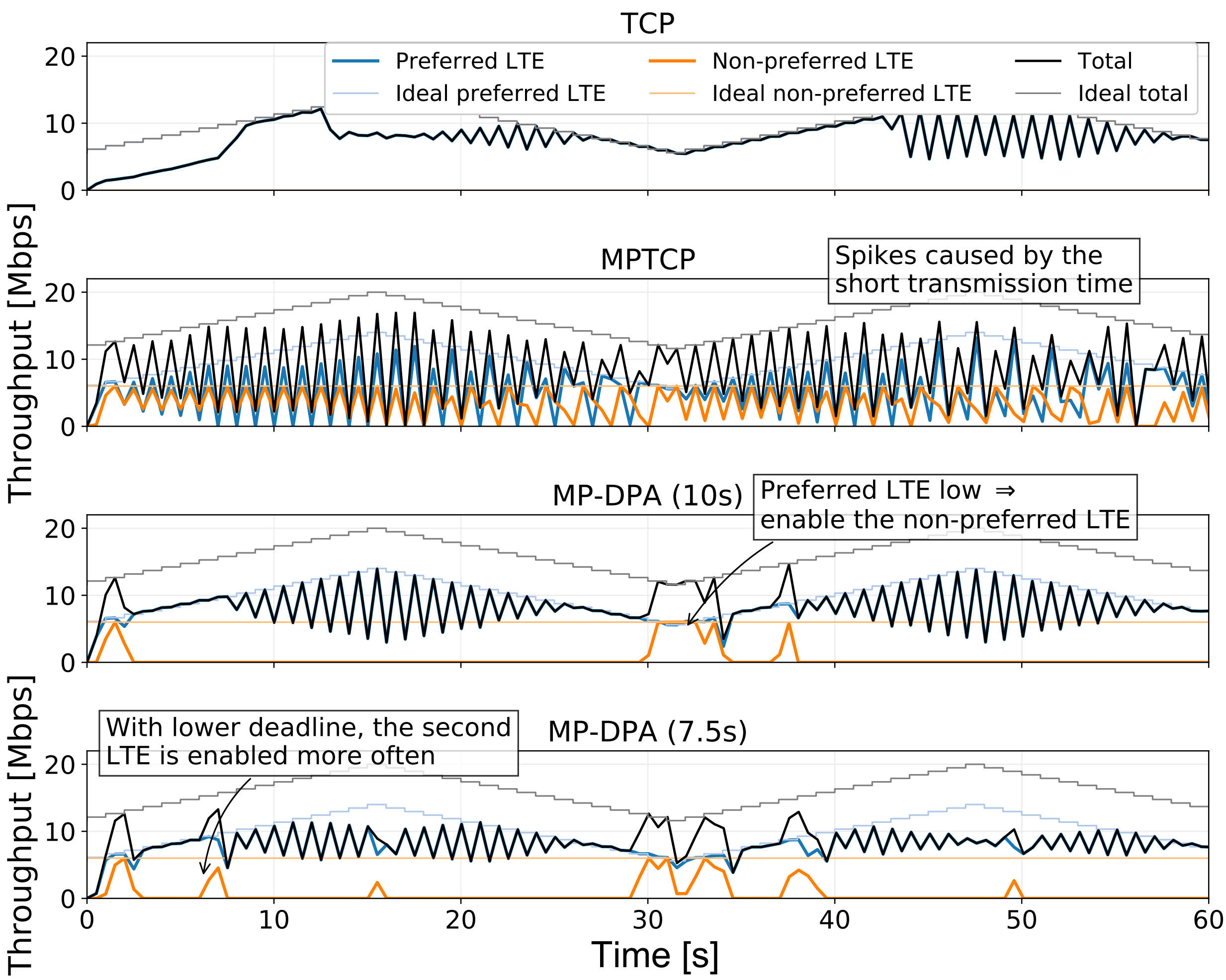


Figure 6.28: Effective throughput during a streaming upload of 60 packets in pseudo-dynamic conditions using TCP (top), MPTCP (middle), and MP-DPA (bottom) with a deadline of 1 and 0.8 seconds. For every packet, MPTCP fully exploits both interfaces to reach the maximum speed. MP-DPA instead, uses the non-preferred LTE only close to the local minimums of the preferred one. With a lower deadline, the second LTE is enabled more often.

viduated also in the streaming upload. If a deadline is more strict, MP-DPA is still able to improve its transmission speed at the expense of more data sent over the non-preferred interface.

6.6.2 Non-Preferred Interface Usage and Reliability Measurements

With the previous considerations, it has proven that, also in a streaming upload context, MP-DPA can adapt its communication performances depending on the deadline value. However, the new MP-DPA is designed as a deadline and preferences-aware protocol to reduce the non-preferred interface usage while respecting deadlines. To validate if MP-DPA meets its specifications with a streaming upload, in the following are analyzed the data distribution among both interfaces and the number of deadlines missed.

Figure 6.30 shows the data distribution using TCP, MPTCP, and MP-DPA for different deadlines.

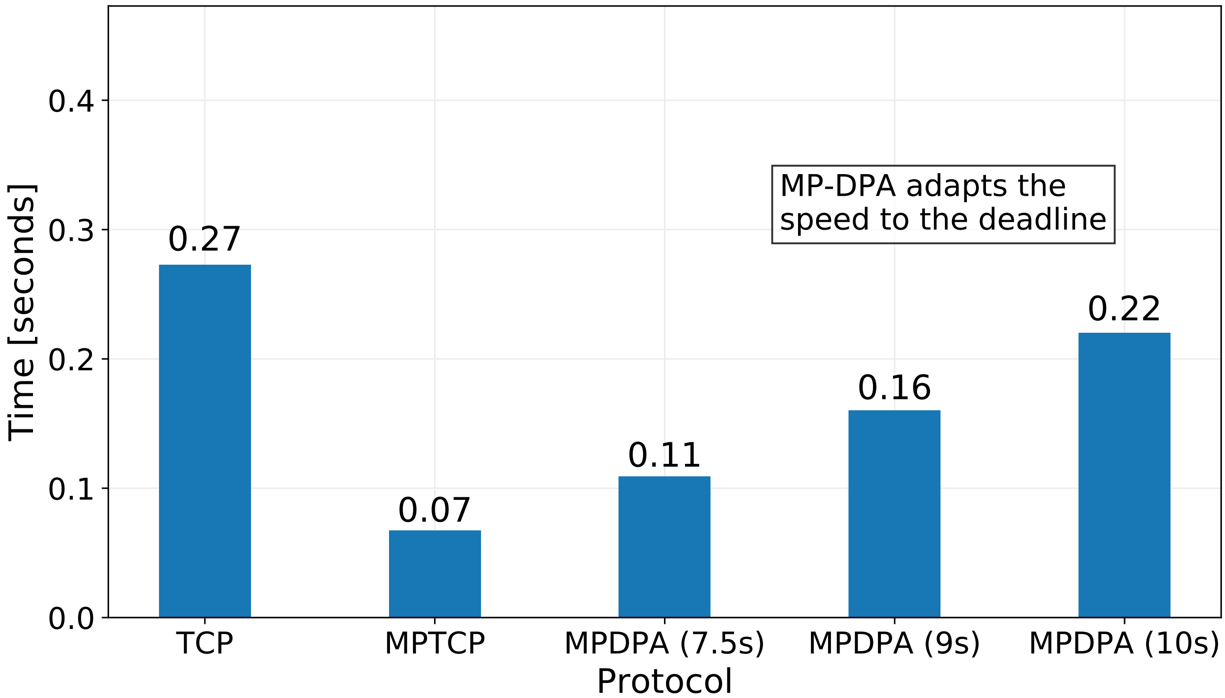


Figure 6.29: Average time needed to sent a packet of 10 MB with TCP, MPTCP and MP-DPA with different deadlines for a streaming upload in pseudo-dynamic conditions. A multilink transmission reduces considerably the upload time. TCP and MPTCP show the same upload time, while MP-DPA adapts its speed on the deadline value.

The highlighted part of the bars is the data sent over the non-preferred interface. In the same way as the bulk upload scenario, MPTCP schedules the packets among the interfaces regardless of the deadline value. For 3 different deadlines, always ∼40% of the data are sent over the second interface. MP-DPA cuts down considerably the usage of the non-preferred interface: from ∼44% of savings for a deadline of 10 seconds to ∼42% with a lower deadline. Comparing this graph to Figure 6.8, in this particular bandwidth configuration, MP-DPA achieves better results which do not decrease substantially for lower deadlines.

The reliability of each protocol expressed as the percentage of deadline missed is plotted in Figure 6.31. Since for most of the triangular shape, the reachable throughput is enough to sustain all the packets transmission, for MPTCP and MP-DPA all the deadlines are respected independently from the value of α. However, in TCP there are still packets that are not completely sent before the deadline expires. In the case of multipath solutions such as MPTCP and MP-DPA, all the packets are entirely uploaded. Since the obtained results are satisfactory and all the deadlines are respected, there is no need to use a lower α value in MP-DPA.

In general, the previous results demonstrate that in a pseudo-dynamic context, MP-DPA behavior is very similar to the bulk upload. Thus, also in a non-already-explored scenario such as the streaming upload, MP-DPA can be considered a valid protocol that can optimize standard multipath protocols.

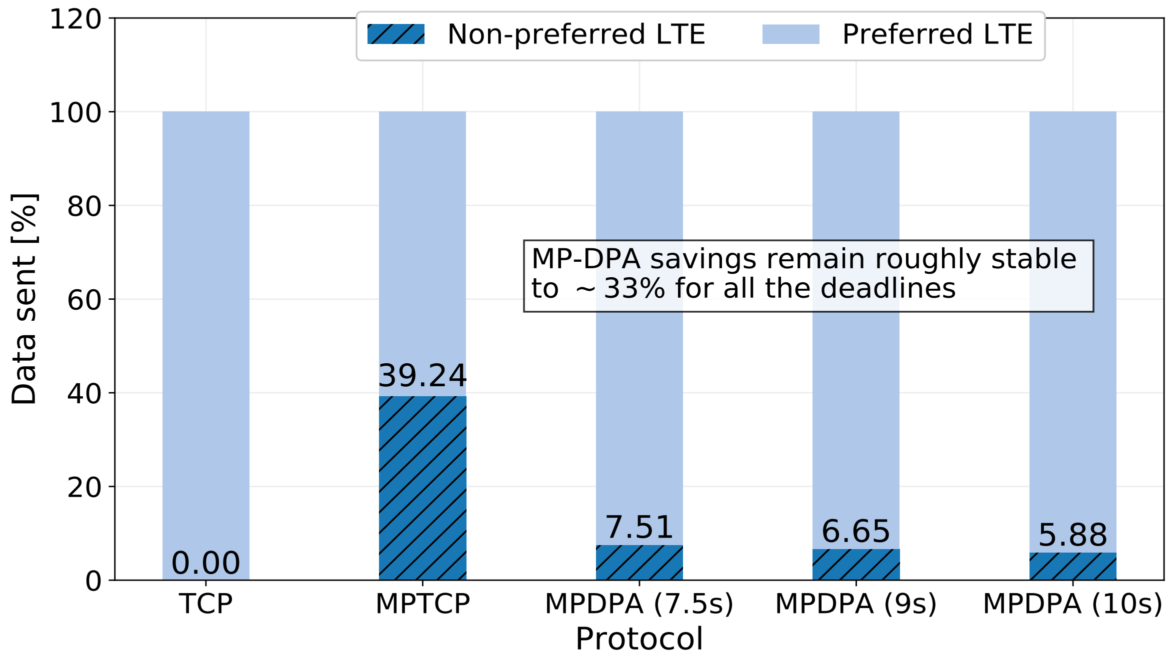


Figure 6.30: Data distribution for TCP, MPTCP and MP-DPA with different deadlines during a streaming transmission of 60 MB of data in pseudo-dynamic conditions. MP-DPA can reduce considerably the usage of the second LTE interface. Contrarily to the bulk upload, the savings do not decrease substantially if the deadline decreases.

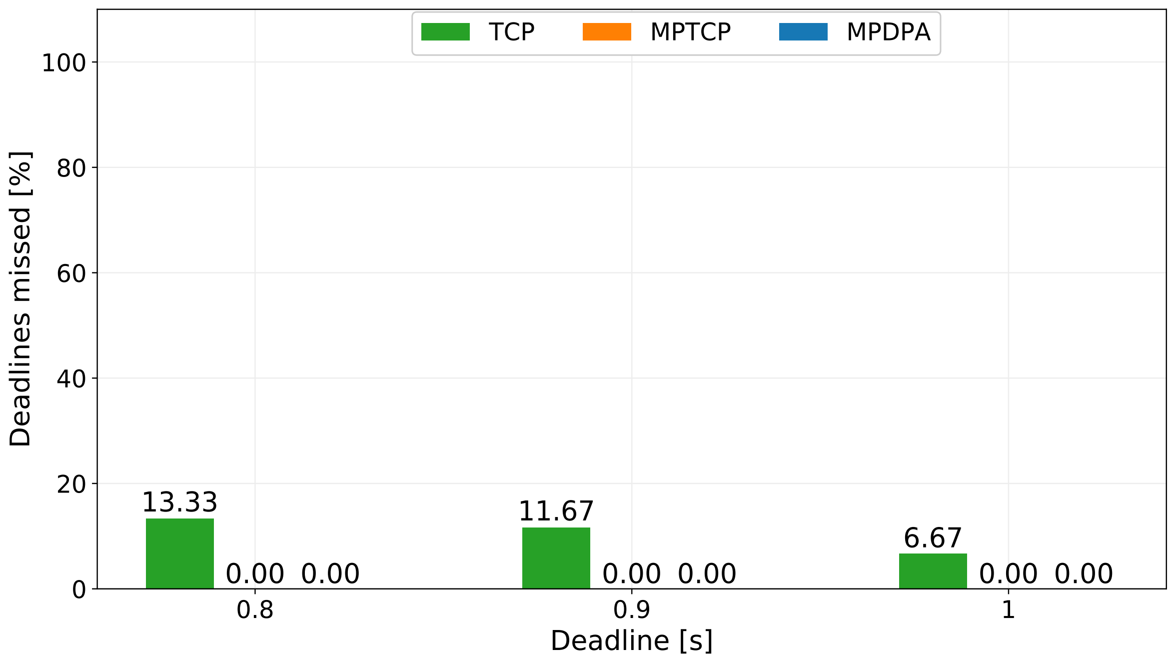


Figure 6.31: Reliability analysis in terms of deadlines missed using TCP, MPTCP, and MP-DPA with different deadlines. A total of 60 packets are sent with streaming upload in pseudo-dynamic conditions. Differently from TCP, MPTCP and MP-DPA are able to sustain the transmission of all the packets.

6.7 Trace-Driven Simulation

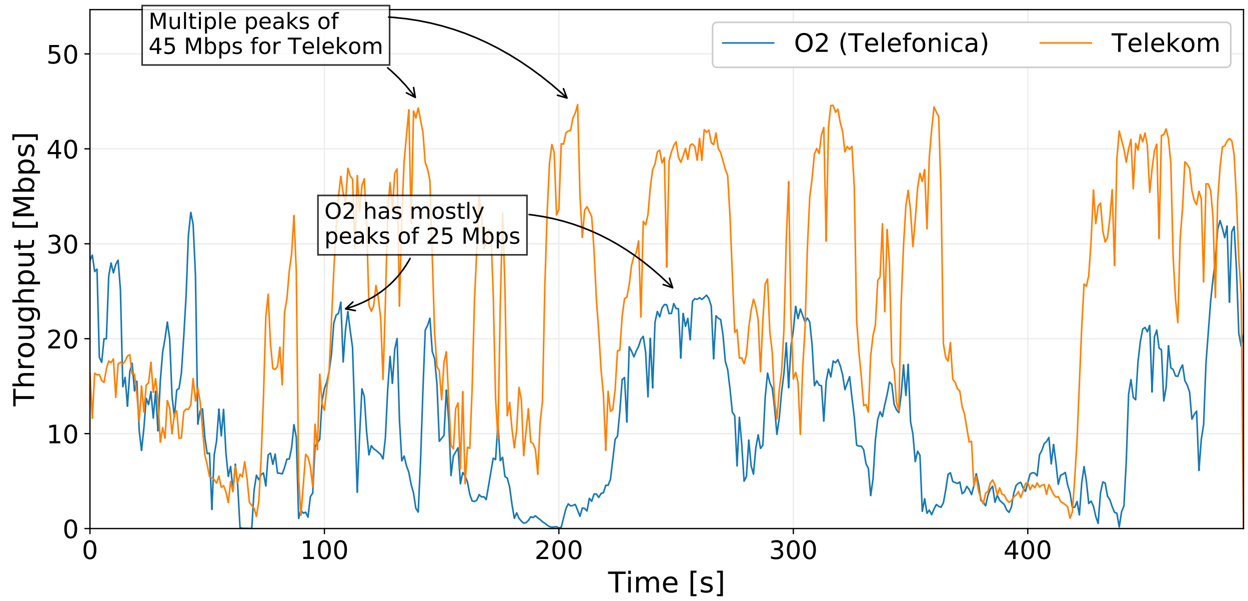
Till now, has been demonstrated that MP-DPA, for both bulk and streaming data upload, can optimize the standard multipath protocols in terms of data savings over the non-preferred interface, while maintaining high performances and reliability in static or pseudo-dynamic conditions.
Unfortunately, in the real-world, bandwidths are not fixed and MP-DPA advantages may vary on the type of environment. Thus, the new communication framework is evaluated in dynamic conditions for highway, rural, and city locations. Also is this case, due to the unpredictability of the bandwidths and according to the conclusions about μ in 6.5.3, the deadline sensitivity α is set to 0.8.
In the next sections are reported firstly the results obtained with the standard vehicle sensor data or bulk upload. In the seconds part of the discussion, the same considerations are made for a streaming upload.

6.7.1 Vehicle Sensor Data Upload

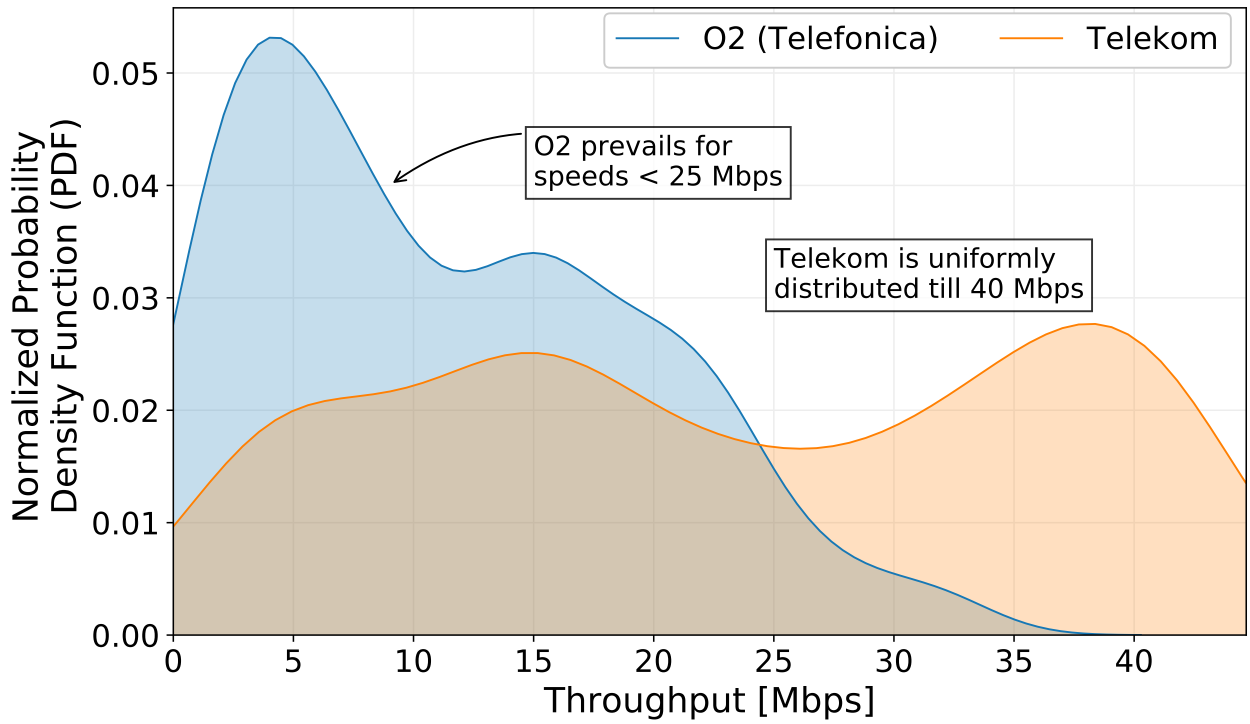
Since a bulk upload is emulated, the following results refer to the transmission of several packets of 100 MB that are sent every 60 seconds.

Highway Trace Results

The obtained measurements obtained with an iperf3 experiment are reported in Figure 6.32. On the upper part of the figure are plotted the traces of both network providers. As introduced before, Telekom has higher performances than O2 for the most part of the simulation. The first reaches in multiple time intervals peaks of $\sim$45 Mbps while the latest can achieve approximately the half.
The speed differences can be more easily detected in the probability density function depicted in Figure 6.32 **(b)**. O2 is predominant in the first part of the graph for lower speeds, while Telekom is uniformly spread among all the values. Despite the differences and some time intervals where both interfaces are low, the total achievable speed with a multilink protocol is quite high.
During the simulation, the preferred and non-preferred interface bandwidths are set according to O2 and Telekom values, respectively. Figure 6.33 shows the effective throughput for MPTCP and MP-DPA.
The recurring behavior of MPTCP can be again identified. Both interfaces are fully exploited to achieve maximum performances. The bottom part of the figure shows the solution proposed by MP-DPA. Contrarily to the preferred LTE that is fully exploited for the whole transmission time, the non-preferred one is enabled only when needed. Each peak represents the moment in which the sender recognizes that the first interface is no more able to sustain alone the transmission of a packet (Algorithm 1). In this way, the other interface is enabled or disabled depending on the goodput estimation in a given time.
In terms of data savings, Figure 6.35 shows the percentages of data distribution among the preferred LTE / O2 and the non-preferred LTE / Telekom of MPTCP and MP-DPA for different deadlines. As expected, MPTCP shows a symmetric behavior for all the deadlines, reaching a relative high value of non-preferred interface usage of $\sim$70%.



(a) Dynamic traces of O2 and Telekom obtained through a 480 seconds iperf test. Telekom often doubles the performances of O2 reaching peaks of 45 Mbps.



(b) O2 is predominant for lower speeds, while Telekom is uniformly distributed from 0 to 40 Mbps.

Figure 6.32: Throughput **(a)** and probability density function **(b)** for O2 and Telekom in highway scenario

Analyzing Figure 6.32 **(b)**, can be deduced that the preferred LTE can sustain alone the transmission of 100 MB in 60 seconds approximately only for the 38% of the simulation time. For this reason, the non-preferred interface is inevitably enabled. However, Figure 6.34 highlights that MP-DPA cuts down the usage of Telekom: ∼10% of savings for 60 seconds of deadline that decreases

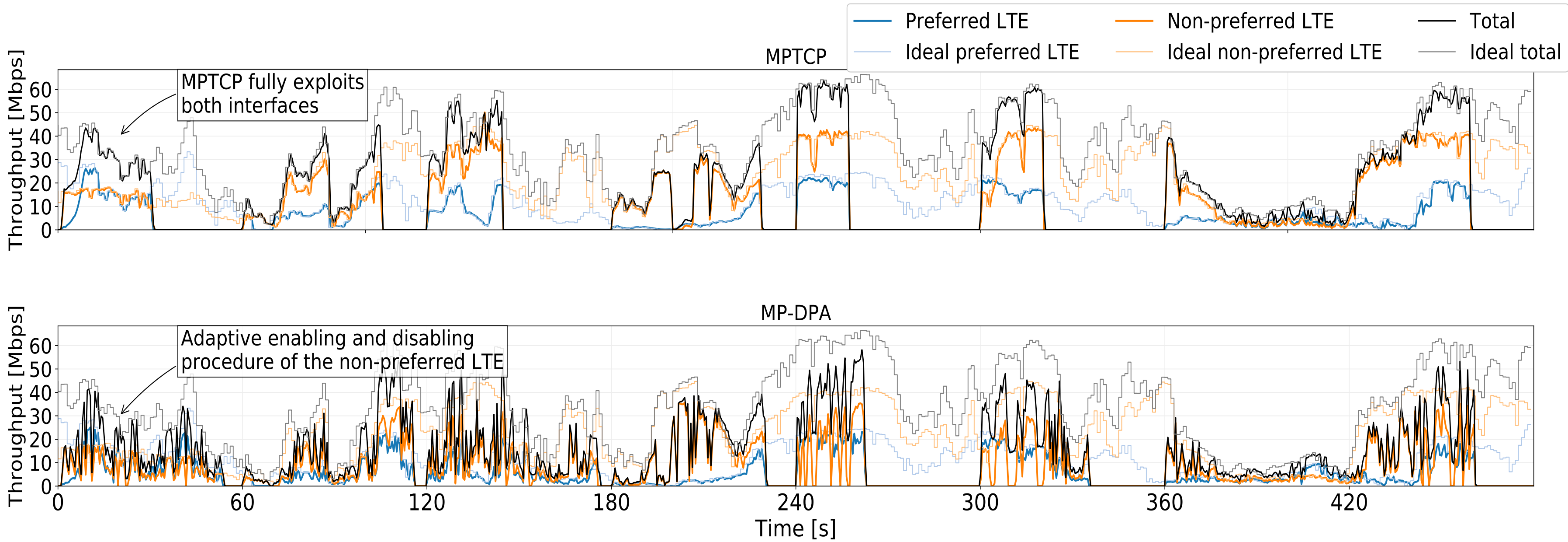


Figure 6.33: Effective throughput during a bulk upload for MPTCP (top) and MP-DPA (bottom) in highway scenario. O2 is assigned to the preferred LTE and Telekom is assigned to the non-preferred LTE. MPTCP leverages all the available bandwidth, while MP-DPA minimizes the usage of Telekom (orange).

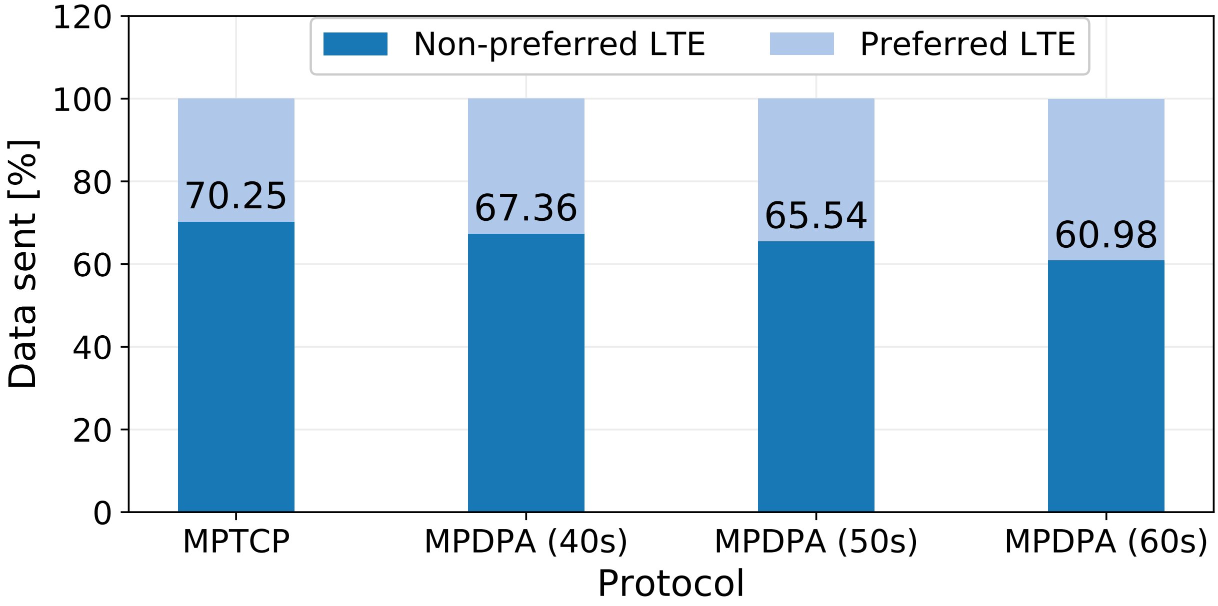


Figure 6.34: Data distribution for MPTCP and MP-DPA with different deadlines during a bulk transmission of 800 MB of data in highway scenario. Using a soft deadline, MP-DPA saves 10% of data over the non-preferred interface. The savings are reduced to 3% for lower deadlines.

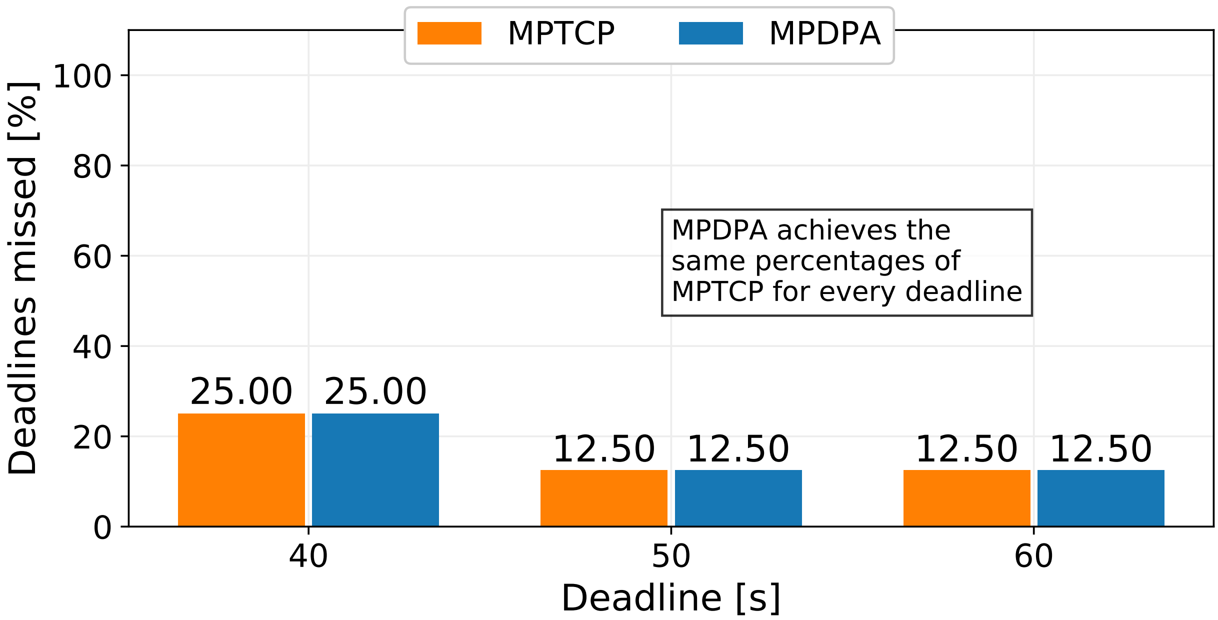
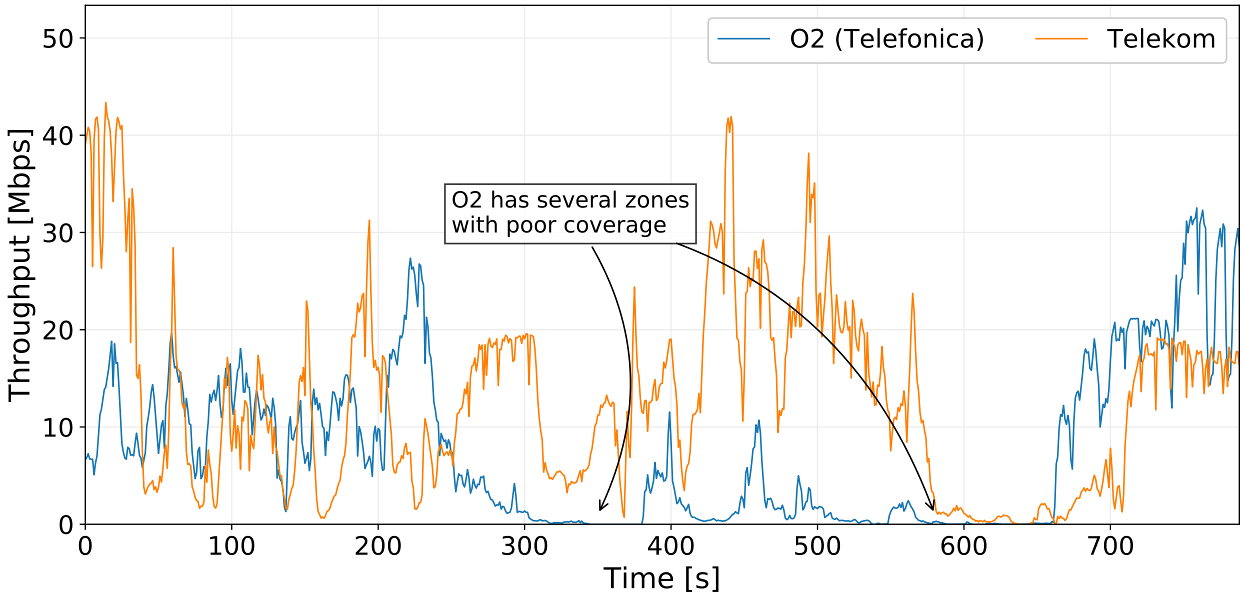


Figure 6.35: Comparing the deadlines missed using MPTCP and MP-DPA with different deadlines in highway scenario. A total of 8 packets are sent with bulk upload every 60 seconds. Contrarily to static experiments, MPTCP does not show a reliable transmission. However, MP-DPA can reach the same percentages obtainable with a multipath protocol.

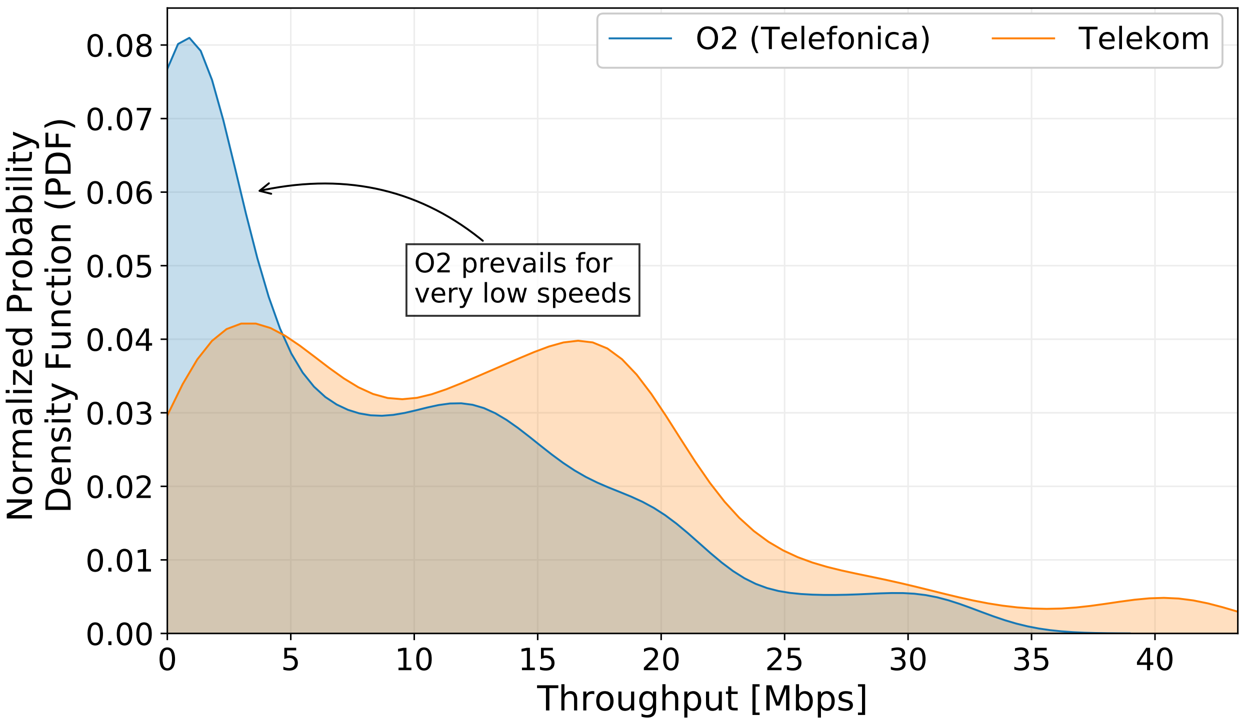
to ∼3% with a lower deadline.
In Figure 6.35, is plotted the reliability of each protocol expressed as the percentage of deadline missed. Being the simulation of 8 minutes, 8 packets must be ideally sent. Since the bandwidths can vary during time, the results strongly depend on the network. Specifically, the highway scenario is characterized by few intervals where the total throughput is not able to sustain the transmission of 100 MB (seconds 360 to 420 in Figure 6.33). MPTCP and MP-DPA percentages are equals: for a relative soft deadline, only 1 packet is not completely sent that increased to 2 with a deadline of 40 seconds.

Rural Trace Results

The obtained traces displayed in Figure 6.36 have completely different values with respect the highway scenario. In case of Telekom provider, there are still high peaks, but it shows numerous bandwidth changes that can rarely reach only dozen of Kbps.
As **(b)** highlights, O2 behaves differently. Its probability distribution function is concentrated mainly for lower speeds. The main aspect that characterized rural experiments is the presence of not covered zones by O2: it will hardly influence the performances and main metrics of MP-DPA as commented later.
As before, in the simulation phase, the preferred and non-preferred LTEs are configured with O2 and Telekom, respectively. Despite the highway and rural traces diversity, MPTCP, and MP-DPA behavior concerning the effective throughput remains the same. Figure 6.37 displays the obtained throughput. Also if the network conditions are not advantageous for MP-DPA, the estimation algorithm triggers the sender to enables the non-preferred interface only when needed.
To deeply understand MP-DPA performances also in not favorable conditions, the most important parameters are taken into account. Figure 6.38 represents the usage of O2 and Telekom interfaces for MPTCP and MP-DPA during the rural traces simulation for multiple deadlines values. The first one exhibits a ∼68% of non-preferred LTE usage while MP-DPA percentages are slightly lower. With small computations, in the current rural context only for ∼22% of the trace, O2 can sustain alone the transmission of every packet. For the remaining simulation time, the second LTE must be enabled. Compared to MPTCP, as the percentages show, the effectiveness of MP-DPA is not satisfying: using a decreasing deadline the savings go from 6% to 1%.
As expected, also the reliability of both protocols is not characterized by satisfactory values as shown in Figure 6.39. In this case, the results refer to 13 packets of 100 MB since the simulation lasts for 13 minutes. Also for a relaxed deadline of 60 seconds, still ∼30% of packets are not entirely transmitted. This value increases to 9 packets over 13 for a deadline of 40 seconds.
Despite these negative results, MP-DPA has the same percentages of MPTCP that represents the maximum reachable performances. This means that, although the connection is poor, MP-DPA does not worsen the MPTCP performance, but tries in every way to optimize the total cost of communication.



(a) Dynamic traces of O2 and Telekom obtained through a 780 seconds iperf test. Telekom and O2 are globally characterized by poor performances: time intervals of good throughput are alternated with worse ones. O2 sometimes shows better performances than Telekom.



(b) O2 is predominant for very low speeds, while Telekom can reach, in average, better values close to 20 Mbps.

Figure 6.36: Throughput **(a)** and probability density function **(b)** for O2 and Telekom in rural scenario.

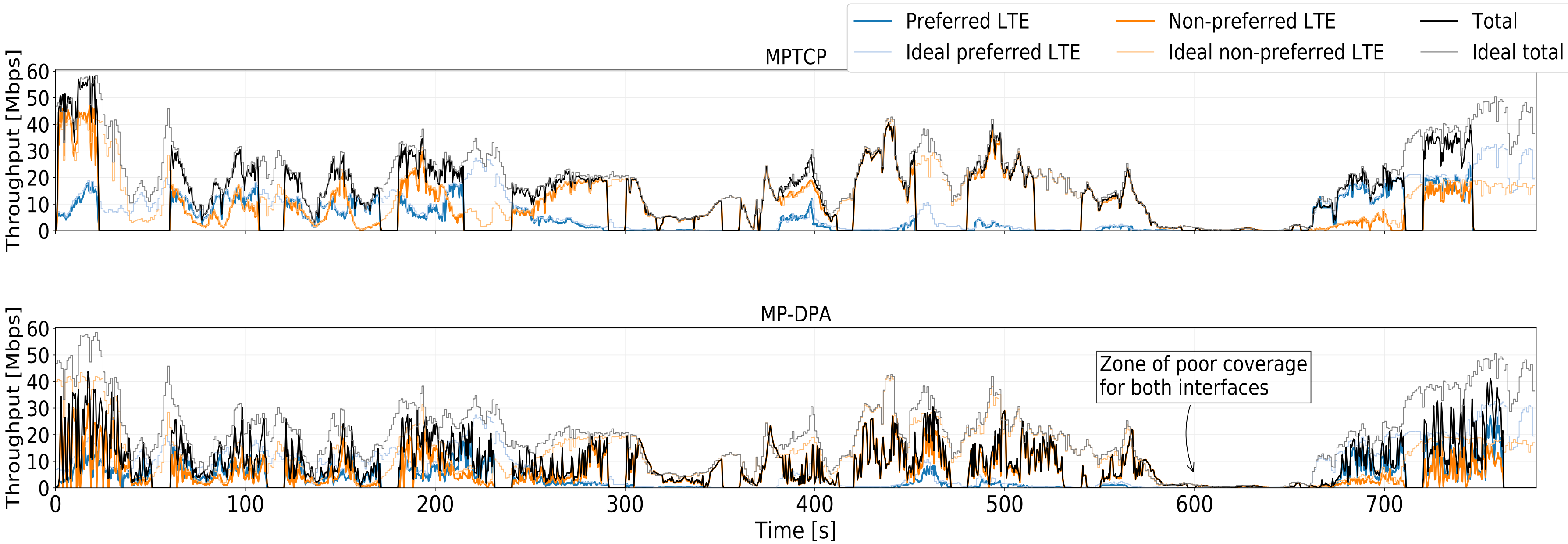


Figure 6.37: Effective throughput during a bulk upload for MPTCP (top) and MP-DPA (bottom) in rural scenario. O2 is assigned to the preferred LTE and Telekom is assigned to the non-preferred LTE. MPTCP leverages both interfaces, while MP-DPA limits the usage of Telekom.

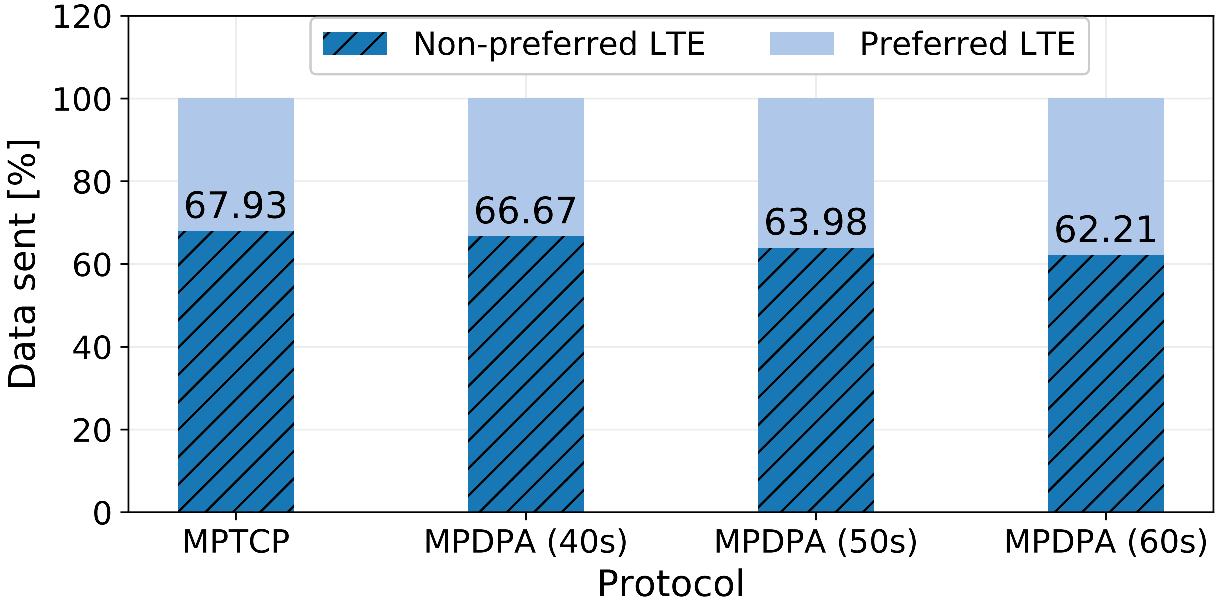


Figure 6.38: Data distribution for MPTCP and MP-DPA with different deadlines during a bulk transmission of 1300 MB of data in rural scenario. Using a soft deadline, MP-DPA saves 5% of data over the non-preferred interface. The savings are near to be negligible for lower deadlines.

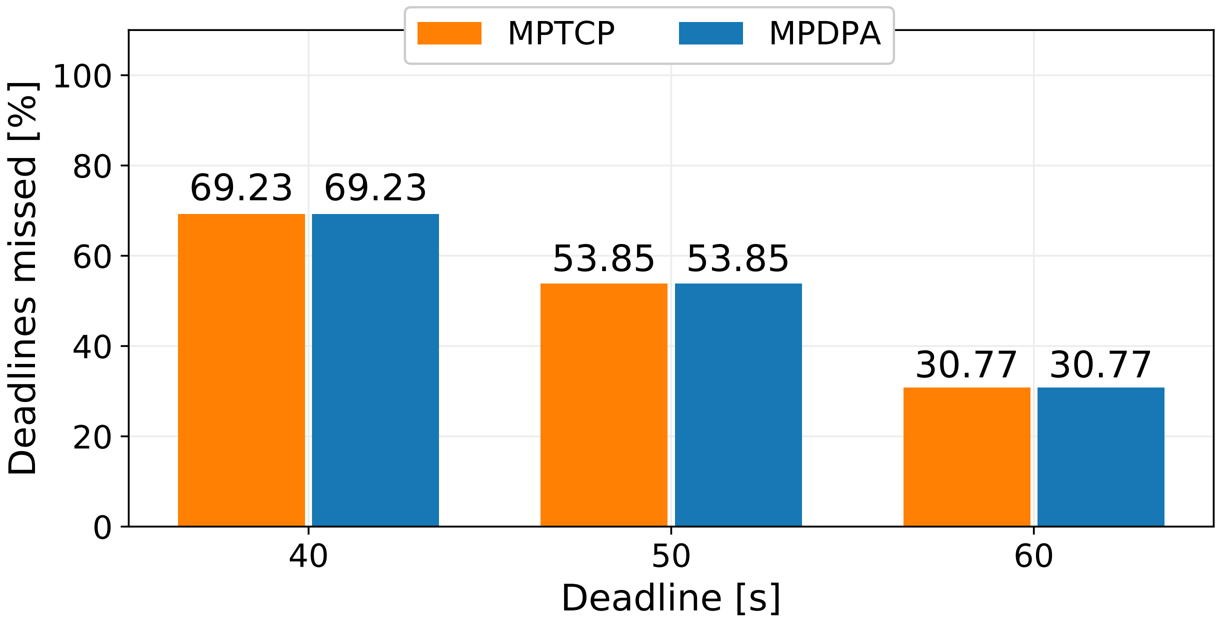
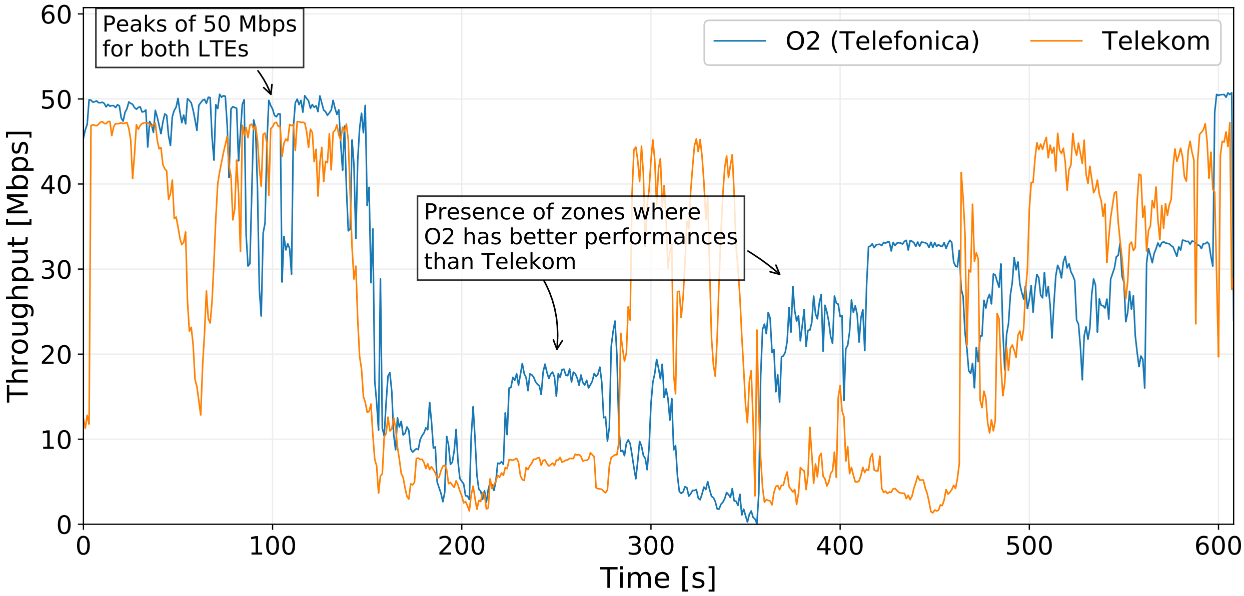


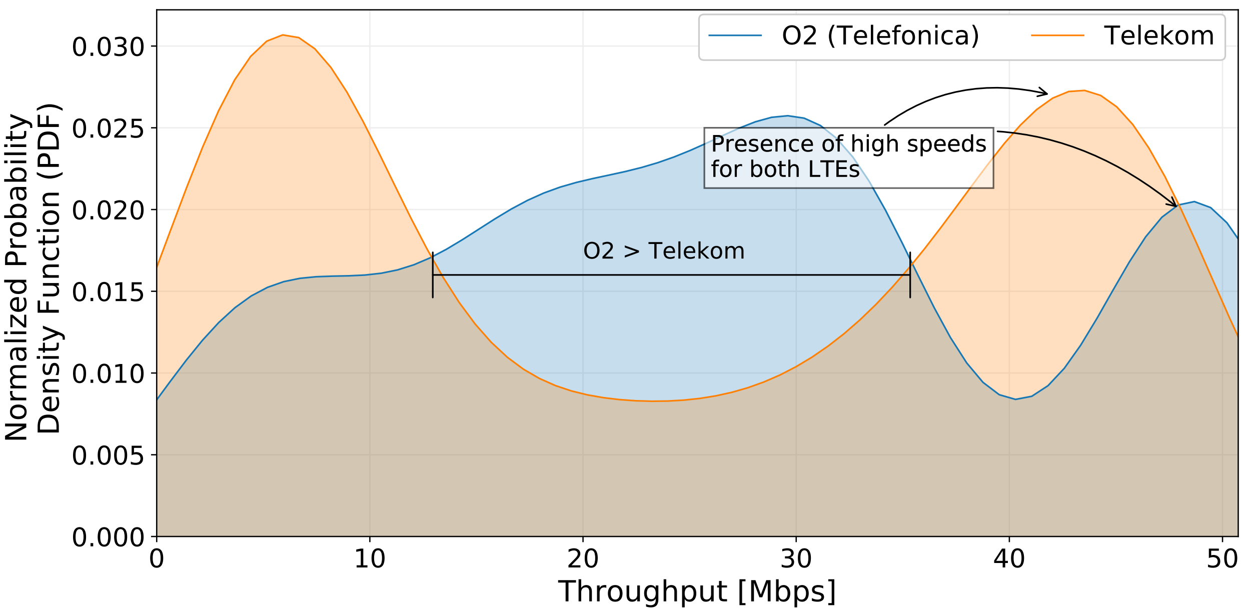
Figure 6.39: Comparing the deadlines missed using MPTCP and MP-DPA with different deadlines in rural scenario. A total of 13 packets are sent with bulk upload every 60 seconds. MPTCP is far from being a reliable protocol: more that half of the packets are not fully sent. However, MP-DPA can reach the same percentages obtainable with a multipath protocol.

City Trace Results

During the experiments for a city location, O2 and Telekom traces are obtained through a simulation of 10 minutes and then plotted in Figure 6.40 **(a)**. O2, which is often considered weak in



(a) Dynamic traces of O2 and Telekom obtained through a 600 seconds iperf test. Telekom and O2 are globally characterized by high performances where peaks of 50 MB can be reach by both. However, time intervals of good throughput are alternated with modest ones. O2 can be sometimes considered better to Telekom.



(b) For both O2 and Telekom, their PDF has not a defined shape. Telekom shows mainly low and high speeds, while O2 is concentrated in the area close to 25 Mbps.

Figure 6.40: Throughput **(a)** and probability density function **(b)** for O2 and Telekom in city scenario

performances, it presents similar bandwidths to Telekom. Here, the reached speed values largely exceed the highway ones specially for O2 that can reach peaks of 50 Mbps.
Is interesting to notice the fact that no network carrier prevails over the other, as happened in the other scenarios. There are time intervals where O2 has higher bandwidths than Telekom and vice versa.
The probability distribution function of bandwidths for both network providers is represented on the bottom part of Figure 6.40. Again, both providers are characterized by variable but overall high bandwidths.

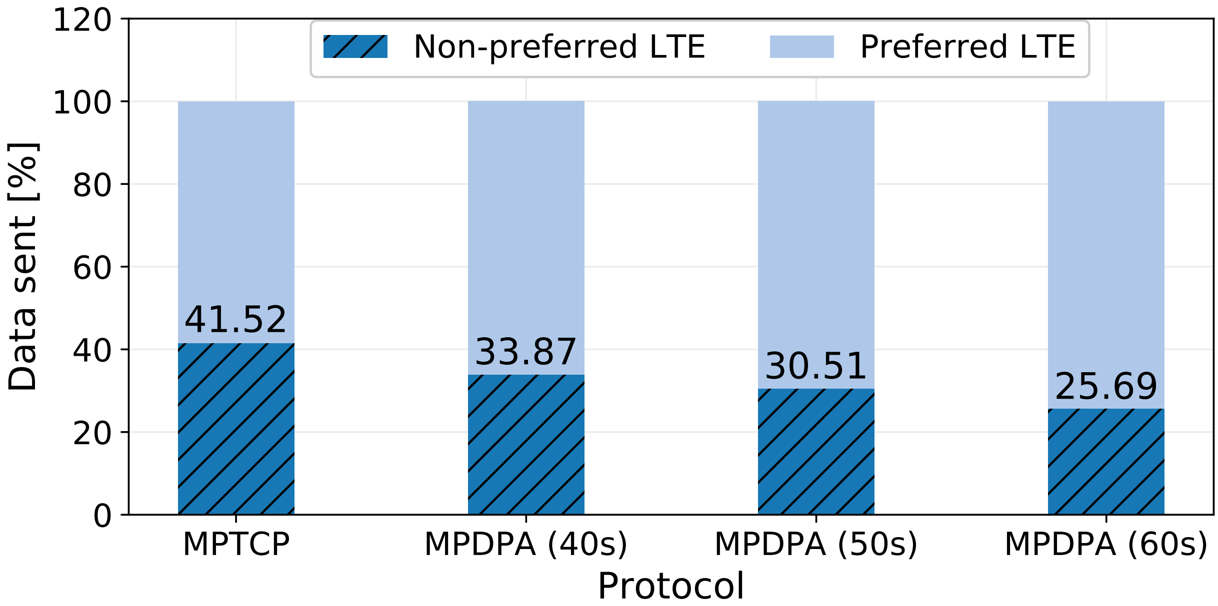


Figure 6.41: Data distribution for MPTCP and MP-DPA with different deadlines during a bulk transmission of 1000 MB of data in city scenario. Using a soft deadline MP-DPA saves 16% of data over the non-preferred interface. The savings are reduced to 7% for lower deadlines.

Figure 6.42 represents instead the simulation results using a deadline of 60 seconds in terms of throughput. Also in this case, is assigned the cheap but weak network provider O2 to the preferred interface and the fast but expensive operator Telekom to the non-preferred one. Also here, in contrast to MPTCP, MP-DPA dynamically enables and disables the non-preferred LTE according to the application layer estimations at the sender side.
The city scenario, for the properties discussed before, can be hypothetically the most effective case for MP-DPA. To confirm that, Figure 6.41 plots the percentage of data sent over both interfaces for different deadlines in MPTCP and MP-DPA.
The deadline and preference-aware protocol can reduce data consumption over Telekom by ~16%. If the deadline is reduced to 40 seconds, the savings are reduced to ~7% compared to MPTCP. However, since O2 trace is characterized by a sufficient bandwidth to transfer alone each packet for the 78% of the simulation, also for MPTCP, the non-preferred interface is used for less than half of the total data.

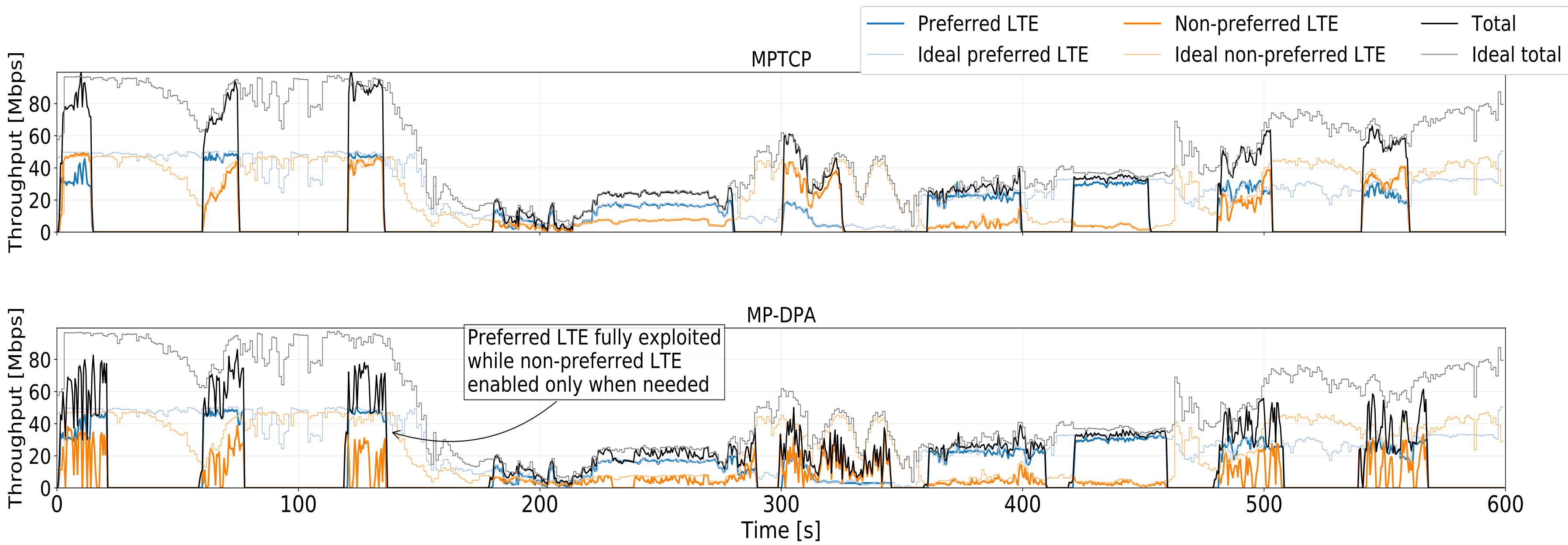


Figure 6.42: Effective throughput during a bulk upload for MPTCP (top) and MP-DPA (bottom) in city scenario. O2 is assigned to the preferred LTE and Telekom is assigned to the non-preferred LTE.

From the reliability point of view, 10 packets of 100 MB each are sent to the receiver with a simulation of 10 minutes. It is important to notice that the city scenario is the only one that can transmit all the packets by respecting each deadline.
Each dynamic trace shows a different behavior compared to the others. MP-DPA performances and advantages are proportional to the bandwidths that the German providers can yield. Although the different results, more or less massive, MP-DPA can optimize standard multipath protocols while maintaining the reliability of the transmission also in dynamic conditions.
As expected, the city scenario is the most advantageous in terms of data savings, followed by the highway and the rural one. To define better the ranking in a dynamic context, the next section analyzes MP-DPA results through the utility factor definition.

MP-DPA Effectiveness Quantification for Bulk Transmission

During the static approach evaluation, a parameter called utility factor μ was used to define a possible pareto-optimal value of α. By knowing the fixed bandwidth of both interfaces, was possible to tune the value of the deadline sensitivity.
Contrarily, in every dynamic trace simulation, since the bandwidth of both interfaces is unknown before the test, α was pre-set to a fixed value of 0.8 (Section 6.5.3). As discussed in 5.1.3, the generic formula used for the utility factor computation is based on the combinations of the two main MP-DPA features and the additional one that considers the overhead of data sent to MPTCP.

In the following, instead of using μ to investigate the best α value, it is exploited as a term of comparison between the different dynamic traces. For this purpose, again, are considered two utility factors: one focuses more on the reliability aspect of MP-DPA while the second one is more centered on the non-preferred LTE/Telekom usage minimization. Table 5.1 resumes the weights value for both utility factors.
To have a better global vision, the results obtained in the simulations of all the dynamic scenarios for different deadline values are reported below in a table format. Table 6.1 summaries them for a deadline of 60 **(a)**, 50 **(b)** and 40 **(c)** seconds. In every table, are reported the results for the utility factor computation: non-preferred LTE / Telekom usage, deadline missed, and data overhead.
Differently from the first two metrics, the data overhead was not deeply investigated. It is computed by analyzing the total amount of data sent with MPTCP and MP-DPA as discussed in 6.5.2.
The higher MP-DPA overhead is caused by the enabling and disabling procedure of the non-preferred LTE / Telekom. As can be noticed in Figure 6.42, MP-DPA throughput is more variable than the MPTCP one. Every time the second interface is switched on and off (for MP-DPA), a small overhead of data is transmitted though it does not belong to the effective payload.
Except for the city location, the deadlines are mostly not respected. If one deadline expires, the sender stops the transmission of the packet. Thus, the total amount of data for highway and rural contexts are globally less than the ones that have to be theoretically sent. For this reason, for highway and rural the data overhead percentage can be omitted for μ computation. However, all of them will be considered to have a correct overview.
The city scenario, having good speeds for O2 and Telekom, can respect all the deadlines. In this

(a) Deadline = 60 seconds.

	Data sent over non-preferred LTE / Telekom	**Missed deadlines**	**Data overhead wrt MPTCP**
Highway	60.9 %	12.5 %	1.1 %
Rural	62.2 %	30.7 %	1.1 %
City	25.7 %	0 %	7.1 %

(b) Deadline = 50 seconds.

	Data sent over non-preferred LTE / Telekom	**Missed deadlines**	**Data overhead wrt MPTCP**
Highway	65.5 %	12.5 %	0.8 %
Rural	63.9 %	53.8 %	1.7 %
City	30.5 %	0 %	5.2 %

(c) Deadline = 40 seconds.

	Data sent over non-preferred LTE / Telekom	**Missed deadlines**	**Data overhead wrt MPTCP**
Highway	66.3 %	25.0 %	1.2 %
Rural	66.7 %	69.2 %	1.7 %
City	33.9 %	0 %	3.0 %

Table 6.1: Summary of MP-DPA main metrics of interest for dynamic experiments in multiple locations during a bulk transmission.

case, the data overhead reproduces the same behavior observed during the static approach: a smaller overhead by lowering the deadline value. From a 7.1% with a deadline of 60 seconds, the amount of data glut is slightly decreased to 3% for a 40 seconds deadline.

For μ_1 computation, MP-DPA effectiveness is evaluated considering the percentage of deadlines missed as metric of major interest. By the combination of the previous results, depending on the network configuration of a given dynamic scenario and the deadline value, MP-DPA has a different impact on the transmission properties. It can be clearly identified in Figure 6.43 where is depicted μ_1 for highway, rural and city scenarios using different deadline values.

The city is the best-case scenario since all the deadlines are respected. Is important to notice that its values are very near to the maximum μ value of 1 that theoretically happens when all the data are sent over O2, the deadlines are always respected and no data overhead is present. However, as in the highway and rural scenario, when the reliability is compromised, μ_1 decreases very rapidly to lower values.

Similar observations can be done for the second utility factor μ_2 where the minimization of Telekom usage is considered as the most relevant metric. The results are shown in Figure 6.44.

μ_2 values are globally smaller compared to μ_1 but the city scenario remains by far the best in terms of MP-DPA advantages. Having the same percentages of data sent over Telekom, highway, and rural shows similar behavior. In the city simulation instead, although μ_2 is reduced by 0.2 points

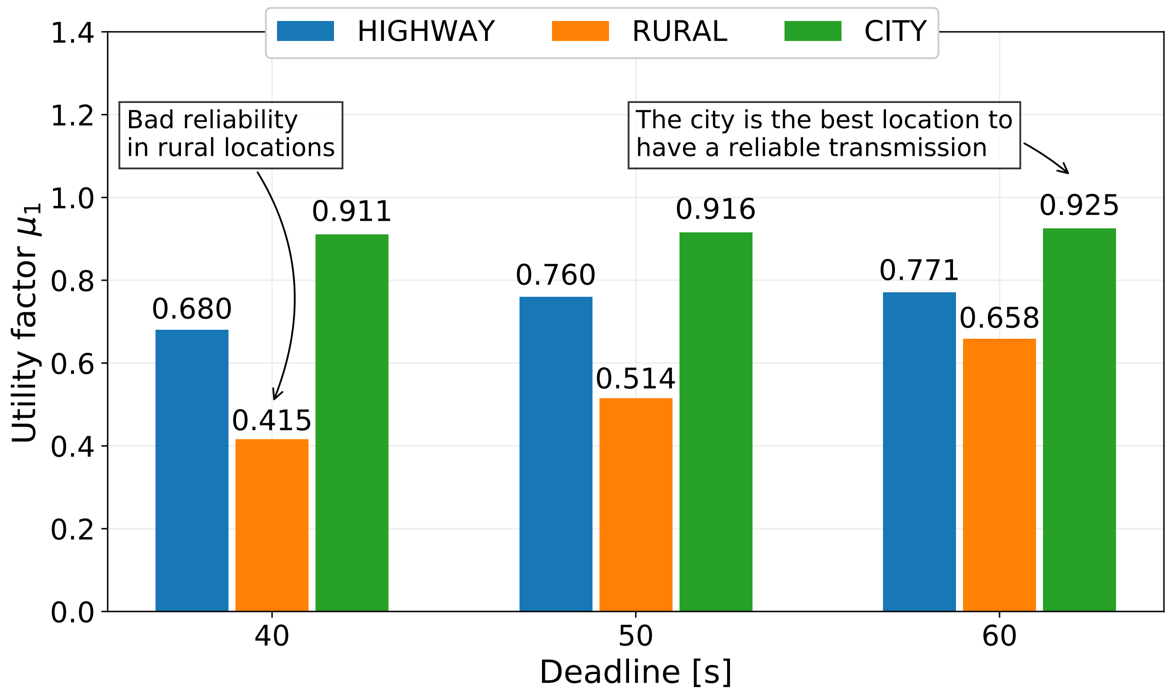


Figure 6.43: Comparing for multiple deadline values the utility factor μ_1 during a bulk upload. In case several deadlines are not respected, the effectiveness of MP-DPA decreases considerably. The city is the best location for reliability purposes.

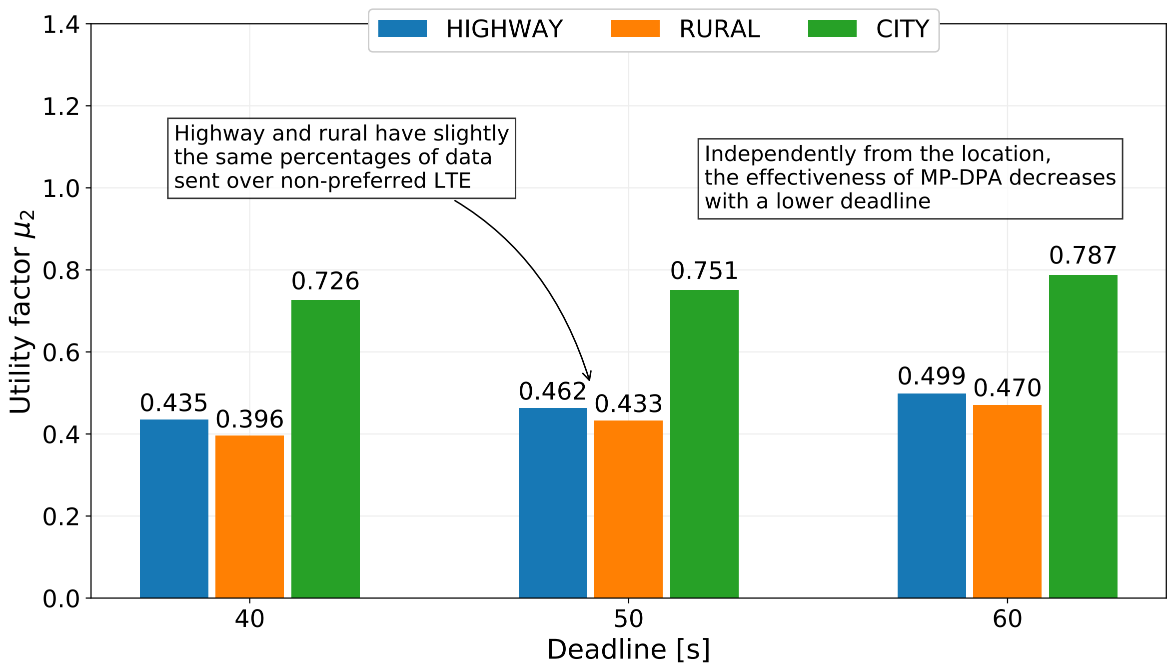


Figure 6.44: Comparing for multiple α values the utility factor μ_2. Since the savings are not satisfactory, unlike the city experiment, μ_2 values are globally lower than μ_1. Thus, city location is still the best scenario for bulk transmissions with MP-DPA.

with respect to μ_1, it presents quite good results.
For all the cases has to notice the fact that, a vehicle provided with MP-DPA protocol can improve the MPTCP transmission cost independently from the type of driving environment. However, a drive experience in the urban scenario can exploit more effectively the advantages that MP-DPA provides. As a consequence, MP-DPA reduced the overall cost of car-to-cloud communication wherever the driven is. Nevertheless, in addition to the network characteristics, MP-DPA results are strongly influenced by other transmission parameters such as the amount of data that has to be transferred and the time interval after which a packet is sent. In the thesis, both were defined and fixed before the simulation phase.

6.7.2 Streaming Upload

Always considering dynamic environments, MP-DPA is also evaluated for a streaming upload as it happens for video streaming cases. From the simulation point of view, the time between the transmission of consecutive packets is smaller than in the bulk scenario. In this case, the transmission is composed of packets of 2 MB that are sent every second.
As discussed in 6.6 for pseudo-dynamic conditions, the way through data is uploaded from car to cloud also influences MP-DPA performances. In this section, MP-DPA is evaluated using the same real-world traces but with a stream-like upload to confirm its effectiveness in a non-already-explored scenario.

Highway Trace Results

Figure 6.45 shows the effective throughput for MPTCP and MP-DPA. Both graphs are characterized by a much more variable throughput than the bulk upload results in Figure 6.33. For every packet, MPTCP tries to fully exploit both interfaces to reach the theoretical maximum bandwidth but practically, it is not achieved due to the short time between two consecutive packets. Before reaching the maximum speed, the first packet is completely sent and the interfaces return to the idle state as long as the next packet is scheduled. Since a high number of packets are sent in a relatively little amount of time, the enable and disable procedure of both interfaces occurs more frequently causing an unstable behavior. Although with smaller peaks, MP-DPA replicates the same MPTCP behavior, but still exploiting less the non-preferred interface.
To test whether MP-DPA meets its specifications, are analyzed the obtained results in terms of data sent on both interfaces and number of deadlines missed. Figure 6.46 shows the percentages of data distribution among preferred LTE / O2 and non-preferred LTE / Telekom of MPTCP and MP-DPA for different deadlines. MPTCP shows a symmetric behavior: for every deadline the second interface is exploited for slightly more than a half of the total amount of data.
Due to the changeable structure of the effective throughput typical of streaming upload, MP-DPA does not significantly reduce MPTCP values. Starting from soft deadlines, the savings amount to only ∼5% that becomes 2% if MP-DPA is subject to stricter deadlines.
In Figure 6.47, is plotted the reliability of each protocol expressed as the percentage of deadline missed. Being the simulation 8 minutes, 480 packets must be ideally sent. MPTCP and MP-DPA percentages are equals: for a relative high deadline, only ∼3% of packets (14 packets) are not completely sent that increased to ∼4% (20 packets) with a deadline of 0.8 seconds.

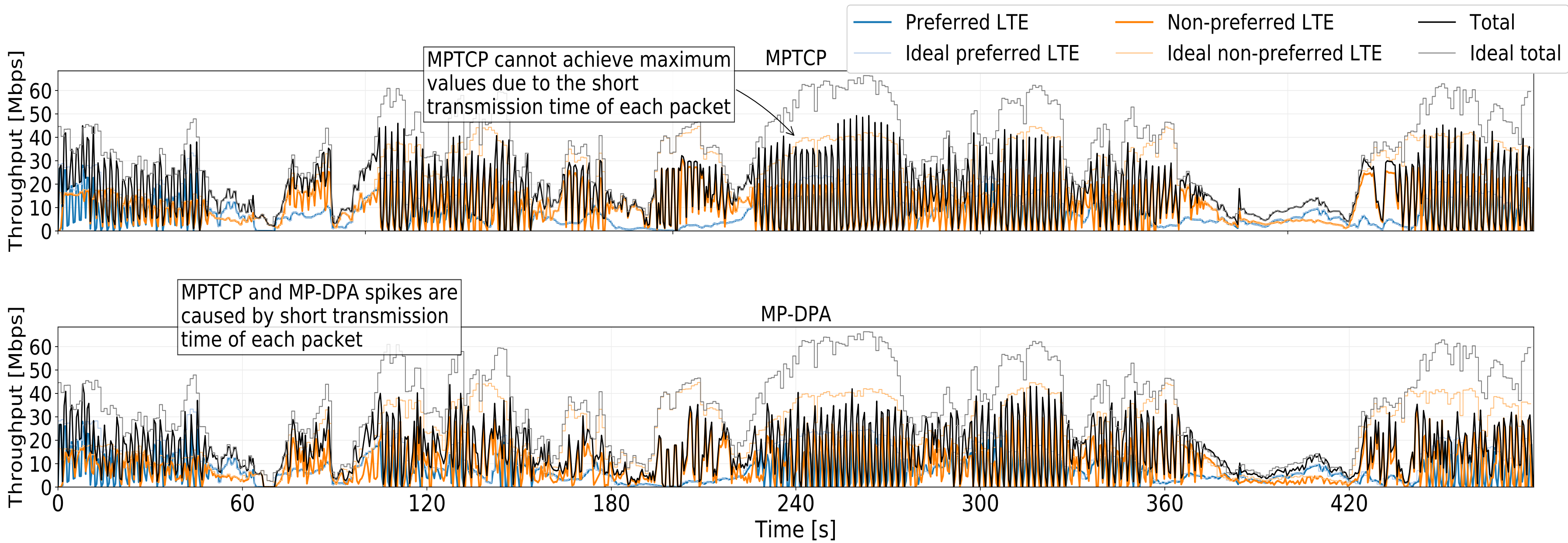


Figure 6.45: Effective throughput during a streaming upload for MPTCP (top) and MP-DPA (bottom) in the highway scenario. O2 is assigned to the preferred LTE and Telekom is assigned to the non-preferred LTE. The spikes are caused by the short time between two consecutive packets. For the same reason, MPTCP cannot fully leverage both interfaces. MP-DPA still minimizes the usage of Telekom (orange).

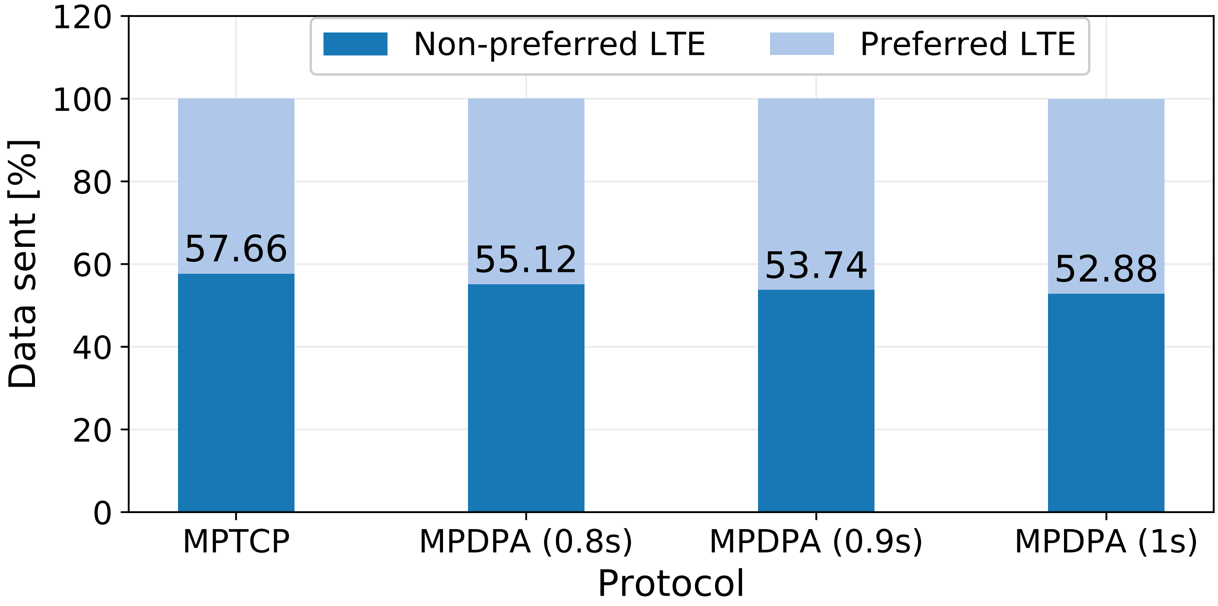


Figure 6.46: Data distribution for MPTCP and MP-DPA with different deadlines during a streaming transmission of 960 MB of data in highway scenario. Using a soft deadline, MP-DPA saves 5% of data over the non-preferred interface. The savings becomes 2.5% for lower deadlines.

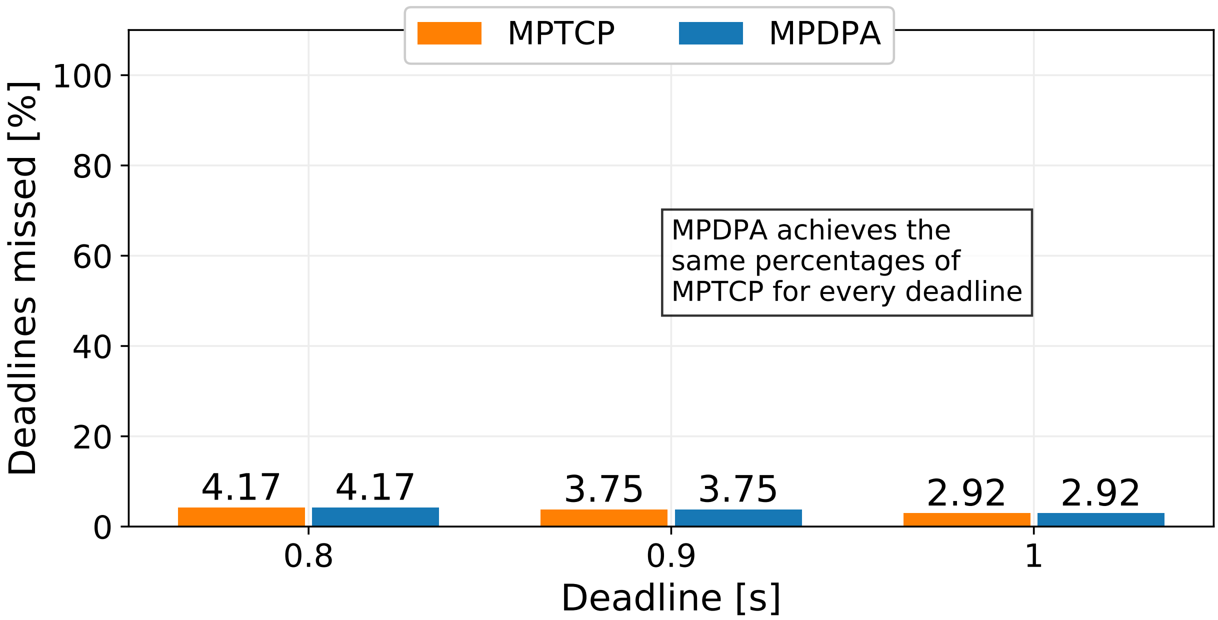


Figure 6.47: Comparing the deadlines missed using MPTCP and MP-DPA with different deadlines in highway scenario. A total of 480 packets are sent with streaming upload every second. MPTCP and MP-DPA show the same relatively low percentages of deadlines missed.

Rural Trace Results

The measurements gathered in Figure 6.36 express that in rural locations the speeds are more concentrated for low values. Moreover, for both O2 and Telekom can be identified time intervals where the network coverage is not sufficient to sustain a reliable transmission.
Despite highway and rural traces diversity, the effective throughput of MPTCP and MP-DPA during simulation displayed in Figure 6.49 remains similar. Also if the network conditions are not advantageous, MP-DPA still limits the used bandwidth of the non-preferred LTE.

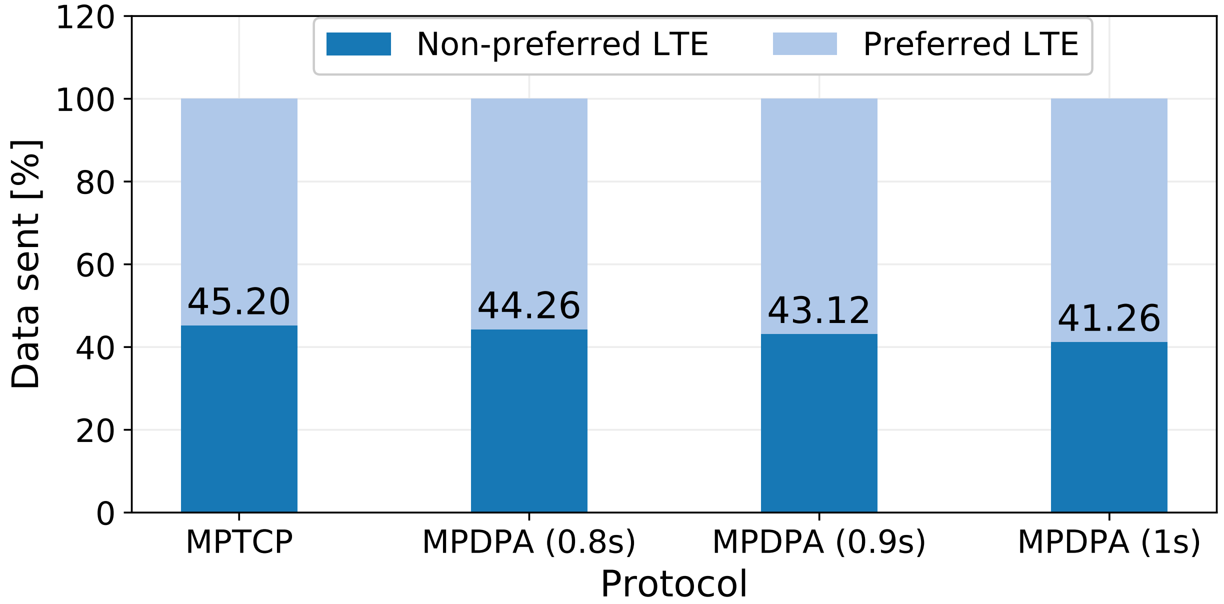


Figure 6.48: Data distribution for MPTCP and MP-DPA with different deadlines during a streaming transmission of 1560 MB of data in rural scenario. Using a soft deadline, MP-DPA saves 4% of data over the non-preferred interface. The savings becomes negligible for lower deadlines.

Since the results obtained in the highway are not considerable, it is expected that in a more non-favorable context like the rural one, MP-DPA does not reach better performances. Figure 6.48 represents the usage of both interfaces during the rural traces simulation for multiple deadlines values. For each deadline, MPTCP schedules ∼45% of data over the Telekom LTE, a value that is nearly achieved also by MP-DPA. As supposed, compared to MPTCP, the effectiveness of MP-DPA in terms of non-preferred LTE usage is not satisfying: using a decreasing deadline the savings decrease from ∼4% to a negligible value.
The reliability of both protocols instead, is characterized by better percentages than bulk upload as shown in Figure 6.50. In this case, the results refer to 780 packets of 2 MB since the simulation lasts for 13 minutes. Differently from the bulk transmission where ∼30% of packets are not sent, here, only ∼9% are non completely sent for a relaxed deadline. Similarly, this value increases to 16.6% compared to 69% of bulk upload with a lower deadline.

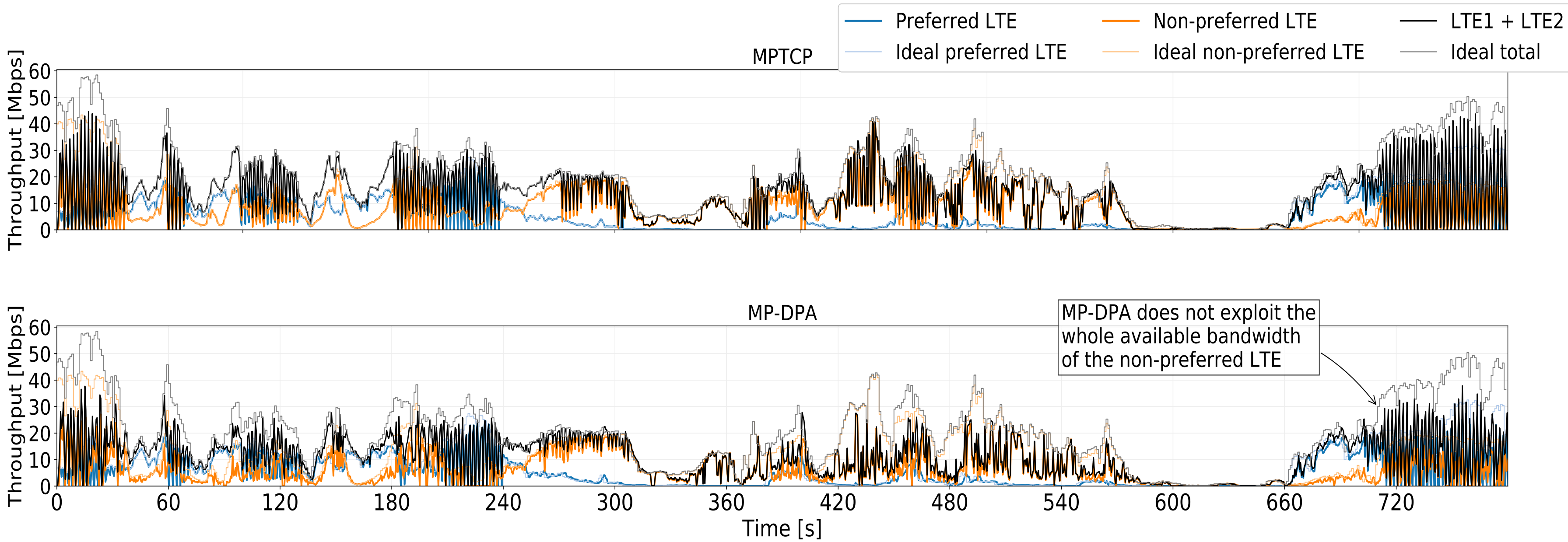


Figure 6.49: Effective throughput during a streaming upload for MPTCP (top) and MP-DPA (bottom) in rural scenario. O2 is assigned to the preferred LTE and Telekom is assigned to the non-preferred LTE. To reduce the data consumption over the non-preferred LTE, MP-DPA does not fully exploits the whole bandwidth of Telekom (orange).

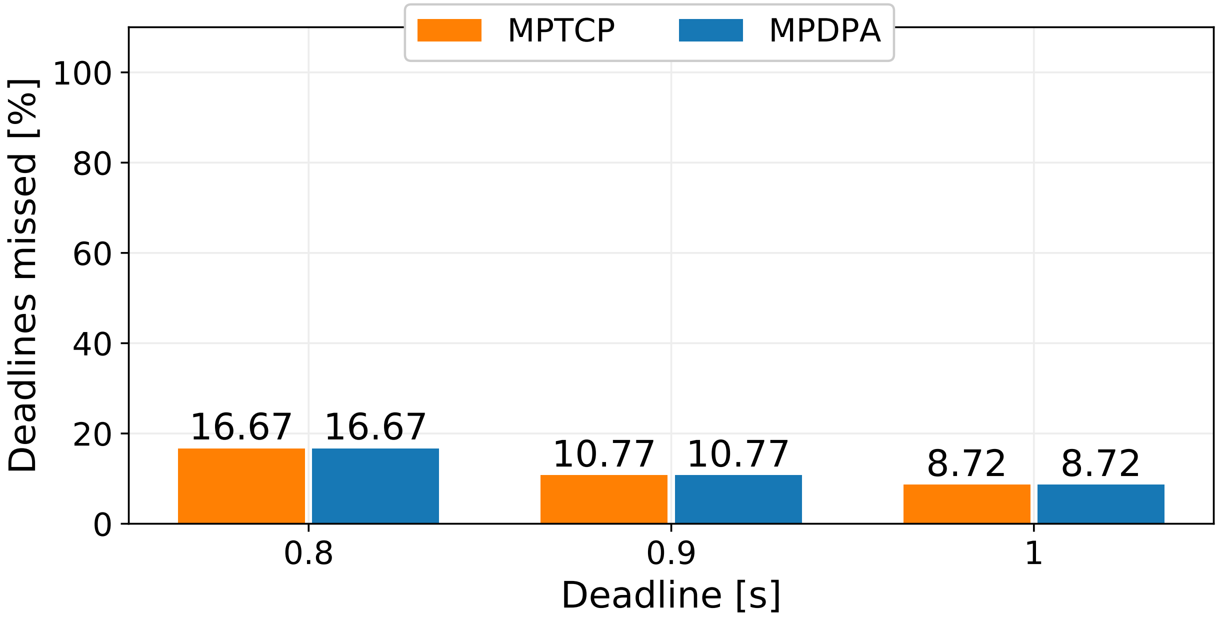


Figure 6.50: Comparing the deadlines missed using MPTCP and MP-DPA with different deadlines in a rural scenario. A total of 780 packets are sent with streaming upload every second. Given the unsatisfying rural performances, several packets are not fully transmitted for both MPTCP and MP-DPA.

City Trace Results

Figure 6.51 represents the simulation results obtained through the city trace. Also here, MP-DPA and MPTCP shows the same high-variable throughput already found in highway and rural streaming results. Both interfaces are dynamically enables and disables due to the short time between two consecutive packets.

The city scenario, for the properties discussed before, can be hypothetically the most effective case for MP-DPA. However, in a stream-like transmission, Figure 6.52 does not depict the expected results in terms of data distribution. Although both interfaces can reach very high values, the deadline and preference-aware protocol is characterized by Telekom data savings values similar to the ones of the other scenarios. MP-DPA reduces the usage of Telekom of ∼7% for a deadline of 1 seconds that decreases to 4% for 0.8 seconds of deadline.

From the reliability point of view, 600 packets of 2 MB each are sent to the receiver with a simulation of 10 minutes. Figure 6.53 highlights the percentages of packet non completely sent. Even if not all deadlines are met as in the case of bulk transmission, in the urban context the transmission in nearly reliable since only 1% of deadlines are missed.

Considering a streaming upload, the analyzed dynamic traces show similar results in terms of data savings and throughput. The reliability instead, reflects the same behavior obtained in the bulk transmission that sees the city as the most reliable scenario, then highway, and last, rural. MP-DPA advantages seem not proportional to the bandwidths that the German providers can yield. Even if the bandwidths are completely different between city, highway, and rural, all the locations have nearly the same percentages of data are sent using Telekom. Although with smaller values, MP-

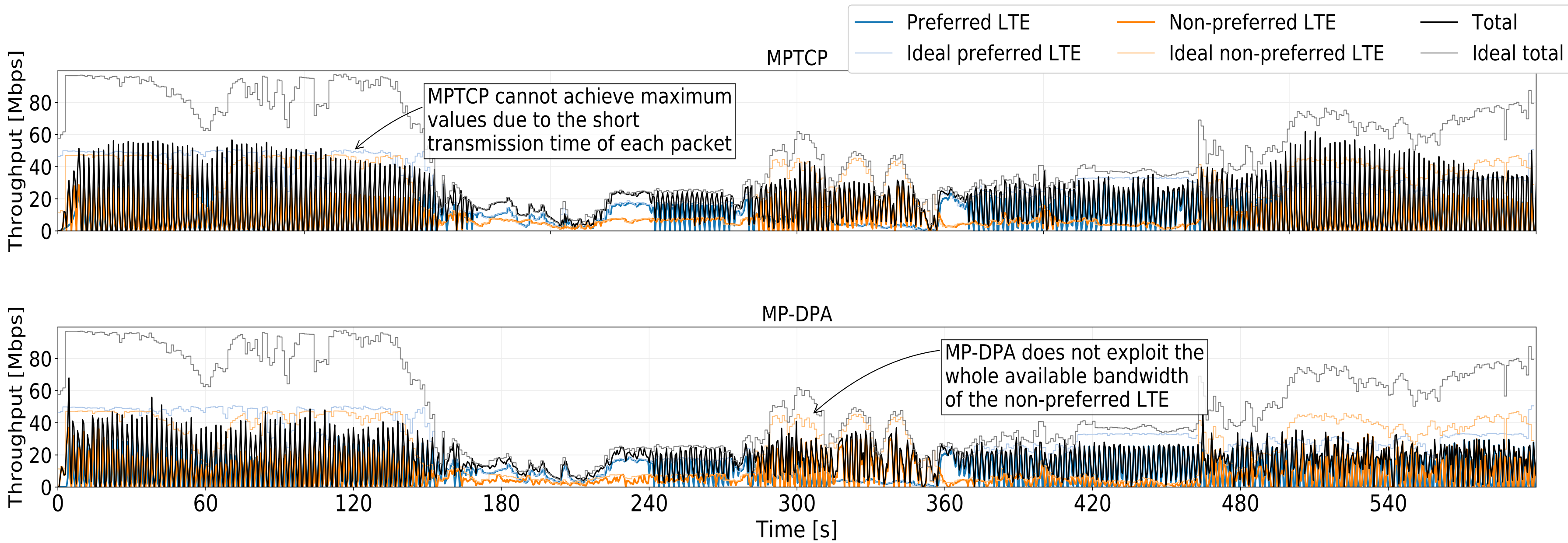


Figure 6.51: Effective throughput during a streaming upload for MPTCP (top) and MP-DPA (bottom) in city scenario. O2 is assigned to the preferred LTE and Telekom is assigned to the non-preferred LTE. The spikes are caused by the short time between two consecutive packets. For the same reason, MPTCP cannot fully leverage both interfaces. MP-DPA still minimizes the usage of Telekom (orange).

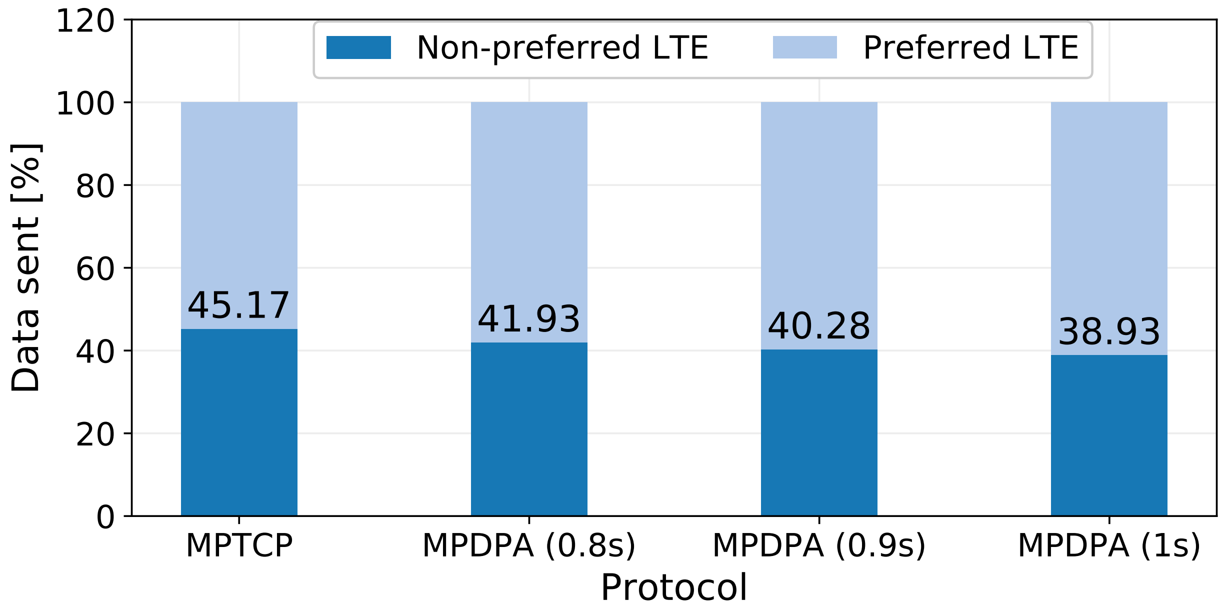


Figure 6.52: Data distribution for MPTCP and MP-DPA with different deadlines during a streaming transmission of 1200 MB of data in city scenario. Using a soft deadline, MP-DPA saves 7% of data over the non-preferred interface. The savings decreases to 4% for lower deadlines.

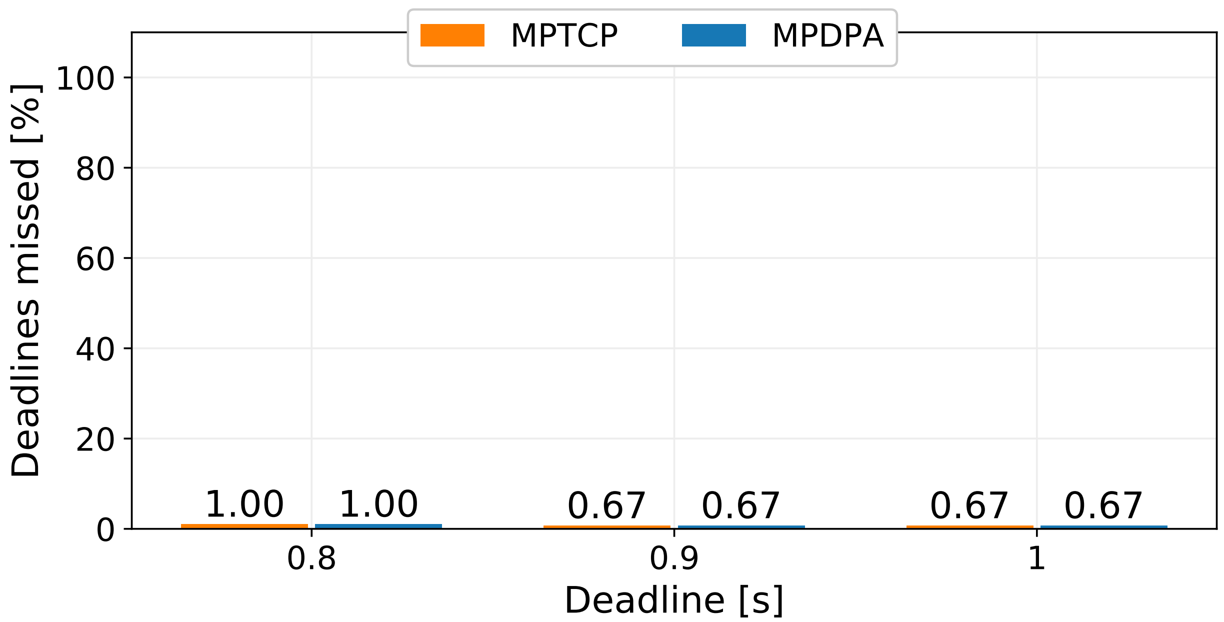


Figure 6.53: Comparing the deadlines missed using MPTCP and MP-DPA with different deadlines in city scenario. A total of 600 packets are sent with streaming upload every second. Although few packets are not fully transmitted, MPTCP and MP-DPA transmission can be considered reliable for each deadline.

DPA still reduces the data usage over the non-preferred interface while showing the same number of deadlines missed of the baseline MPTCP. Thus, it is still valid that MP-DPA can optimize standard multipath protocols while maintaining the reliability for a bulk and streaming transmission.

MP-DPA Effectiveness Quantification for Streaming Transmission

Even if the trace-driven results for a streaming transmission are similar, MP-DPA effectiveness is measured through both utility factors. Firstly, the focus is on the reliability aspect of MP-DPA while successively, it is more centered on the Telekom usage minimization.
The results obtained in the stream-like simulations of all the dynamic scenarios for different deadline values are reported below in a table format. In Table 6.2 are reported the percentages for a deadline of 1 **(a)**, 0.9 **(b)** and 0.8 seconds **(c)**. In every table, are reported the results for the utility factor computation: Telekom usage, deadline missed and data overhead. The data overhead metric,

(a) Deadline = 1 second.

	Data sent over non-preferred LTE / Telekom	**Missed deadlines**	**Data overhead wrt MPTCP**
Highway	52.9 %	2.9 %	0.1 %
Rural	41.3 %	8.7 %	0.1 %
City	38.9 %	0.6 %	0.1 %

(b) Deadline = 0.9 seconds.

	Data sent over non-preferred LTE / Telekom	**Missed deadlines**	**Data overhead wrt MPTCP**
Highway	53.7 %	3.7 %	0.2 %
Rural	43.1 %	10.8 %	∼0 %
City	40.3 %	0.6 %	0.1 %

(c) Deadline = 0.8 seconds.

	Data sent over non-preferred LTE / Telekom	**Missed deadlines**	**Data overhead wrt MPTCP**
Highway	55.1 %	4.1 %	0.2 %
Rural	44.3 %	16.6 %	∼0 %
City	41.9 %	1.0 %	0.1 %

Table 6.2: Summary of MP-DPA main metrics of interest for dynamic experiments in multiple locations during a streaming transmission.

defined as the difference between the total data sent with MPTCP and MP-DPA, presents negligible values. It is motivated by the fact that for MPTCP and MP-DPA, both interfaces are enabled and disabled with nearly the same frequency causing a similar data glut although the protocols are different. Approximately every second, both interfaces are enabled when the packet is sent and disabled at the end of the packet transmission. Therefore, also MPTCP, which in bulk upload holds enabled both interfaces for more time, presents the same amount of data sent of MP-DPA.
For the μ_1 computation, MP-DPA efficacy is evaluated considering the percentages of deadlines

missed as metric of major interest. By the combination of the previous results, MP-DPA has a different impact on the transmission properties. It can be clearly identified in Figure 6.54 where is depicted μ_1 for highway, rural and city scenarios using different deadline values.

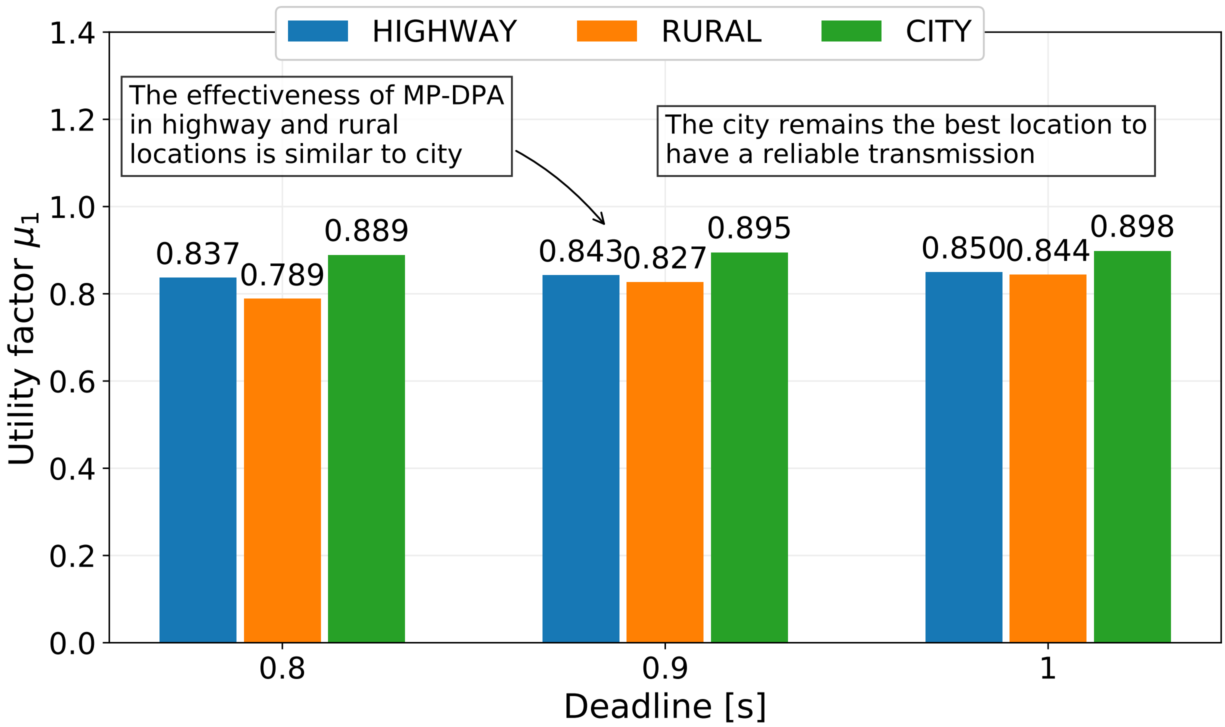


Figure 6.54: Comparing for multiple deadline values the utility factor μ_1 used for reliability purposes during a streaming upload. All the 3 locations have acceptable reliability properties thus, μ_1 values are close to each other and the ideal value 1. Nevertheless, the best location remains the city one.

Although the city scenario does not show important improvements in terms of savings like the bulk upload, it is still the best-case scenario since almost all the deadlines are respected. Is important to notice that in this case, all the values have acceptable reliability properties and thus, are very near to each other and close to the maximum μ value of 1.

For the second utility factor μ_2, where the minimization of Telekom usage is considered as more relevant metric, the values in Figure 6.55 are globally lower than μ_2. Comparing the results obtained in different locations can be noticed a completely different behavior from previous utility factors. The city scenario is not anymore the one that dominates over the others even if it is by far the best in terms of bandwidths. Surprisingly, the rural location, although its bad performances, does not show the worse results since MP-DPA was able to exploits Telekom less frequently than the highway.

The value of the deadline influences directly both μ_1 and μ_2. With a lower deadline, MP-DPA loses in performances but it still provides advantages compared to the baseline MPTCP.

Thus, in the case of a stream-like transmission, wherever the driver is, MP-DPA can reduce the overall cost of car-to-cloud communication. If MP-DPA is used for bulk and streaming upload, the driving experiences in the urban scenario remain the most effective.

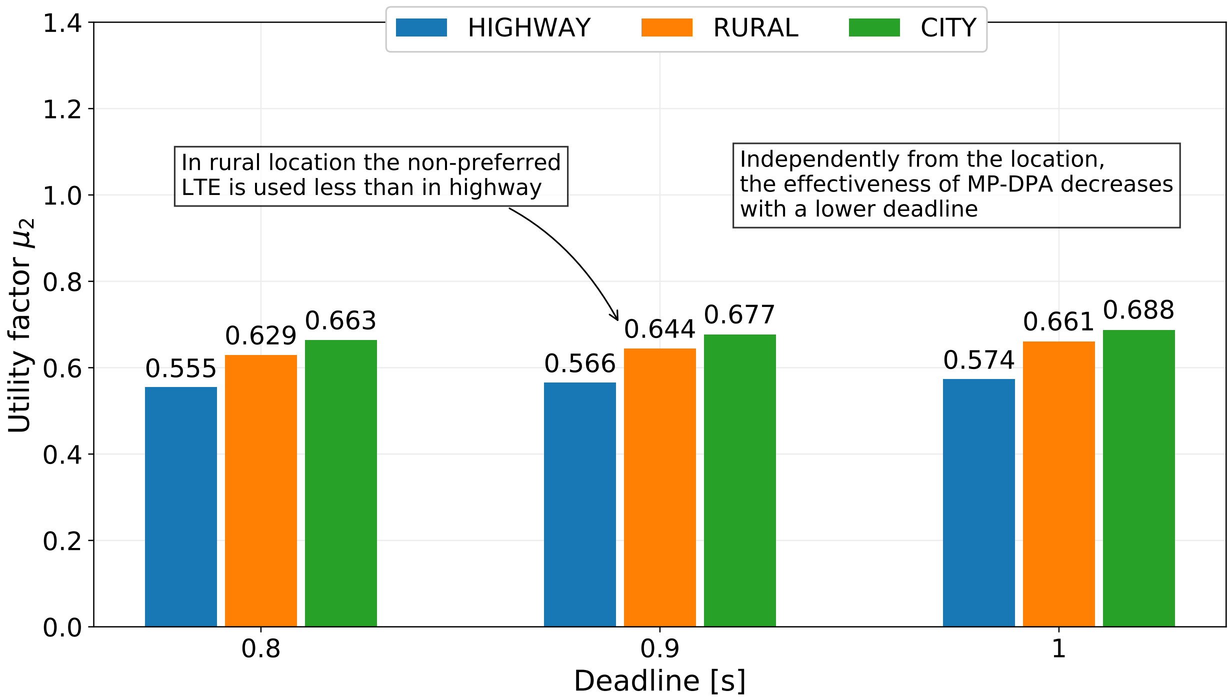


Figure 6.55: Comparing for multiple α values the utility factor μ_2 which mainly focus on the non-preferred LTE usage reduction. Since the savings are not considerable, μ_2 values are globally lower than μ_1. Thus, although the rural location has poor performances, its values are comparable to the city one.

6.8 MP-DPA Performances in Different Locations

In the previous section, the data rate is not stable and the coverage of network providers is not uniform. The already described experiments are conducted at different locations in the North Rhine-Westphalia state in city, highway, and rural locations.

In particular, to evaluate MP-DPA in real conditions, the preferred interface is associated with O2 measurements while Telekom is represented by the non-preferred interface since O2 is considered cheaper than Telekom [48]. For every scenario, MPTCP and MP-DPA are compared in terms of throughput behavior, data sent using Telekom, and deadlines missed.

In terms of data savings, the plots in Figure 6.56 summarize the obtained results for MPTCP and MP-DPA for multiple deadline values during a bulk and stream-like transmission. From a global point of view, MP-DPA values (green and red bars) are always below the MPTCP ones (blue and orange bars). It confirms that a vehicle provided with an MP-DPA protocol can improve the transmission performances with respect one that exploits MPTCP, independently from the type of driving environment and data upload type. As a consequence, MP-DPA reduces the overall cost of car-to-cloud communication wherever the driver is. It is important to notice that MP-DPA, in case of streaming upload (red bars), is not strongly influenced by the network configuration and thus, by the environment. Indeed, the percentages range in relatively small: for every location, the data sent over Telekom are all between 55% and 39%. However, a drive experience in the urban scenario can exploit a slightly more effectively the advantages of MP-DPA.

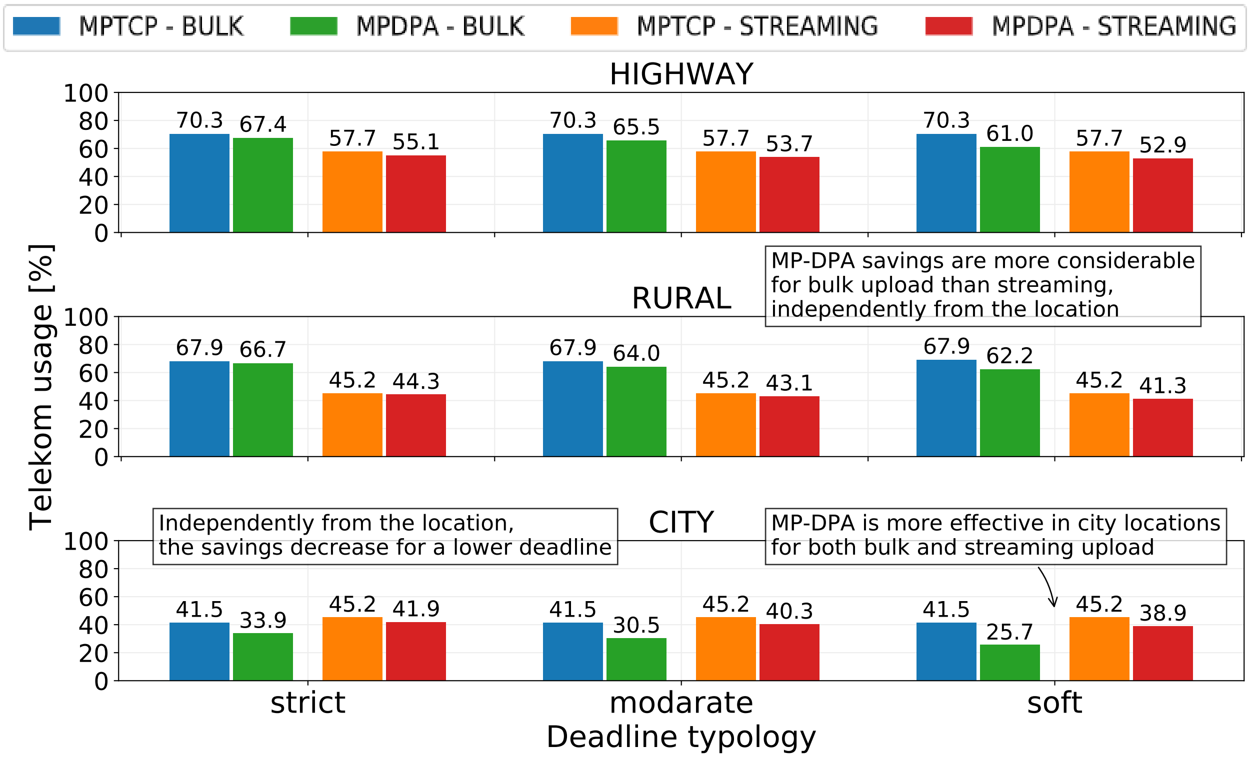


Figure 6.56: Summary of the usage of Telekom for MPTCP and MP-DPA for bulk and stream-like data upload with different types of deadlines. During a bulk transmission, MP-DPA brings considerable advantages to MPTCP, especially for soft deadlines. In a streaming transmission, the range of savings from a soft to a strict deadline is relatively small. However, in both cases, the city location provides the best conditions to make MP-DPA effective.

In the bulk upload (green bars), this phenomenon is more evident than in the stream one. In the city trace, using MP-DPA, from 25% to 34% of the data are sent through Telekom, differently from the other traces which show even 67% of data sent over the non-preferred interface.
Considering the differences between MPTCP and MP-DPA, during a bulk transmission, MP-DPA can provide very good advantages, especially for soft deadlines. In the case of a stream-like transmission instead, MP-DPA and MPTCP have similar percentages of data sent over Telekom. Their values are nearly equivalent to strict deadlines.
From the reliability point of view, Figure 6.57 summarizes the percentages of deadlines missed of MP-DPA for multiple deadlines in all the considered traces for bulk and streaming data upload. There is no distinction between MPTCP and MP-DPA since the latest can respect the same percentages of deadlines as the standard multipath protocol. It means that although the reliability can be compromised, MP-DPA does not worsen the standard multipath performances.
During a bulk data upload, the driving environment strongly influences the reliability of the MP-DPA transmission. In case of a high available bandwidth like the urban trace, all the deadlines are met. With a worsening of the network conditions such as rural, a relatively high number of packets are not completely sent. The same behavior cannot be observed for a stream-like transmission, where the percentages of deadlines missed do not follow the same bulk-like principle. During a

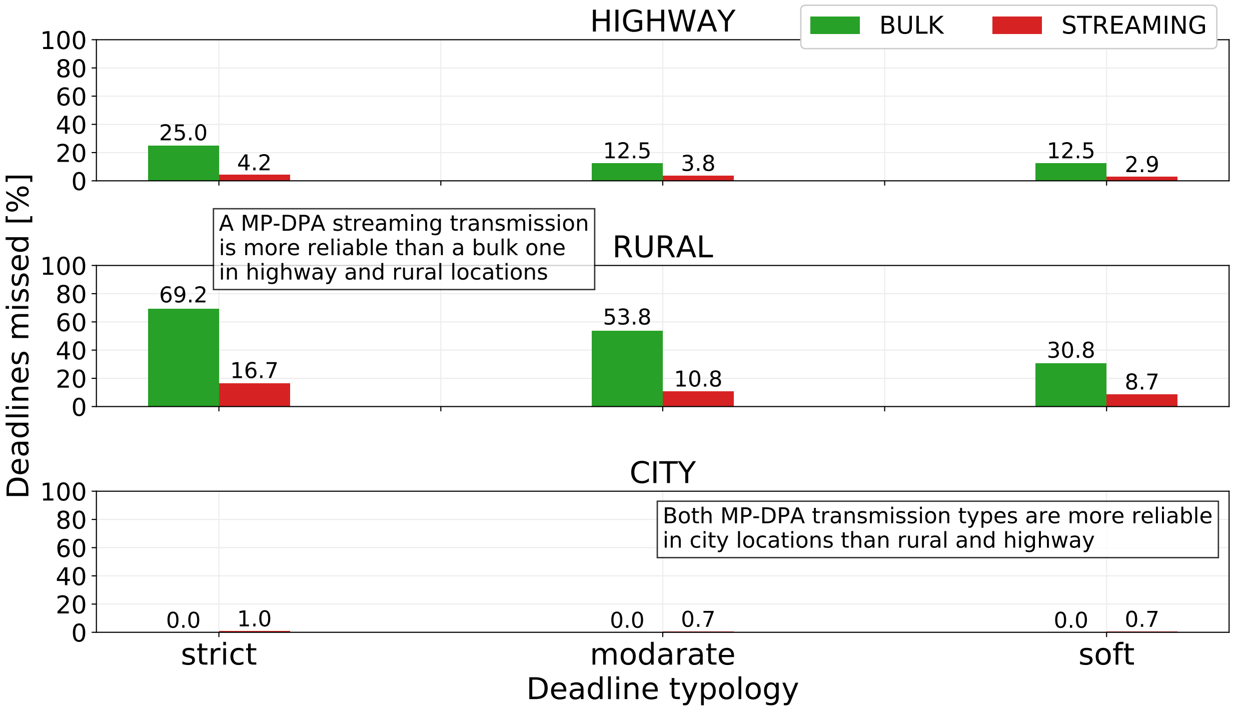


Figure 6.57: Summary of the deadlines missed of MP-DPA for a bulk and streaming upload with different types of deadlines. MP-DPA can maintain the same reliability of a standard multipath protocol. In bulk transmissions, the driving environment strongly influences the reliability of MP-DPA. For a stream-like upload instead, the percentages do not reach higher values but they vary between 0.7% ad 16.7%

streaming upload, the range in which the deadlines missed vary is relatively low: from 0.7% to 16.7%. Contrarily, during a bulk upload, the deadline missed can even reach 70%. Anyway, the urban environment still presents the best values for a streaming transmission although few packets are not completely sent.

7 Conclusion and Future Works

In this thesis, is conducted the first study of preference-aware multipath for adaptive car-to-cloud communication. The implemented Multipath Deadline and Preference Aware (MP-DPA) framework is proposed and instantiated in the context of static and dynamic network conditions. MP-DPA aims to reduce the usage of the non-preferred or most expensive interface to reduce the overall transmission cost while maintaining standard multipath protocol features such as high throughput and reliability. It is achieved by minimizing the scheduling of packets over the non-preferred LTE and exploiting the whole bandwidth of the preferred one.

For evaluation purposes, in addition to standard bulk data upload, MP-DPA performances are also explored for the first time in a stream-like transmission. After standard experiments in favorable and non-favorable conditions, MP-DPA limitations and weaknesses are analyzed for static bandwidths. Further experiments are conducted to investigate a possible pareto-optimal value of α that maximizes MP-DPA advantages in given network configurations.

Through extensive field studies in highway, rural, and city locations in Germany, is demonstrated that MP-DPA enhanced connections are robust and adaptive. Compared with the off-the-shelf MPTCP, MP-DPA minimizes the non-preferred interface data usage without decreasing the reliability, but at the expense of a slight data overhead. However, the new framework performance strongly depends on the type of upload and driving location. In a rural location, defined as the worst-case scenario, compared to the baseline MPTCP, the new MP-DPA reduces the Telekom usage by 6% and 4% with bulk and streaming upload, respectively. The urban area instead is considered as the best scenario in which MP-DPA advantages can be fully exploited with percentages of savings that increased by up to 16%. In addition to the network characteristics, MP-DPA results are strongly influenced by other transmission parameters. For instance, MP-DPA performances depends also on the amount of data that has to be transferred and the time interval after which a packet is sent. For the thesis purposes, both were defined and known before the simulation phase. Since all the dynamic experiments are conducted with a fixed α, the next step is the implementation of an enhanced version of MP-DPA that makes it further adaptive in terms of deadline sensitivity. By making the algorithm able to recognize when is possible to decrease or increase α, MP-DPA can achieve better performance and dynamically adapt the scheduler to the network conditions. Moreover, can be noticed that all the simulations are conducted on a laptop instead of a real vehicle. To be able to get results even closer to the real use of MP-DPA, the new framework can be integrated into a real device (e.g. Raspberry Pi) and then evaluated in real time driving experiments. It is expected that due to the different operating conditions, MP-DPA performances will be slightly different but still valid and advantageous.

List of Abbreviations

CMP Connectivity Management Platforms

LAN Local Area Network

QoS Quality of Service

TCP Transmission Control Protocol

MPTCP Multipath Transmission Control Protocol

MP-DASH Multi Path Dynamic Adaptive Streaming over HTTP

MP-DPA Multipath Deadline and Preference Aware

V2X Vehicle-to-everything

CVIM Common Vehicle Information Model

UAV Unmanned Aerial Vehicle

RTT Round Trip Time

QoE Quality of Experience

LTE Long Term Evolution

WLAN Wireless Local Area Network

BLEST BLocking ESTimation

STTF Shortest Transmission Time First

OLIA Opportunistic Linked-Increases Algorithm

BBR Bottleneck Bandwidth and Round-trip propagation time

ScalaNC Scalable Network Coding

MoSoNets Mobile Social Networks

SPDY Software Package Data Exchange.

HTTP Hyper Text Transfer Protocol

V2V Vehicle-to-Vehicle

V2I Vehicle-to-Infrastructure

V2N Vehicle-to-Network

DSRC Dedicated Short-Range Communications

RSU Road Side Units

3GPP 3rd Generation Partnership Project

C-V2X Cellular V2X

3G Third Generation

4G Fourth Generation

NAT Network Address Translations

IP Internet Protocol

BBR Bottleneck Bandwidth and Round-trip propagation time

OLIA Opportunistic Linked-Increases Algorithm

SINR Signal to Interference and Noise Ratio

CAA Congestion Avoidance Algorithm

IPV4 Internet Protocol Version 4

API Application Programming Interface

Bibliography

[1] Han, Bo and Qian, Feng and Ji, Lusheng and Gopalakrishnan, Vijay, “MP-DASH: Adaptive Video Streaming Over Preference-Aware Multipath,” in *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 129–143. [Online]. Available: https://doi.org/10.1145/2999572.2999606

[2] Wikipedia, “Connected car,” https://en.wikipedia.org/wiki/Connected_car#cite_note-mashconnectedcar-1, 27 March 2020, [Online; accessed 19-April-2020].

[3] Matthias Maierhoferm, “An Overview of Connected Cars and the Automotive industry,” https://www.mavoco.com/overview-connected-cars/, 2. April 2019, [Online; accessed 19-April-2020].

[4] “Automotive Big Data Marketplace for Innovative Cross-sectorial Vehicle Data Services - Project description,” https://automat-project.eu/tags/project-description, 2020, [Online; accessed 19-April-2020].

[5] Wikipedia, “Connected car - Wikipedia, the free encyclopedia,” https://en.wikipedia.org/wiki/Connected_car, 2020, [Online; accessed 19-April-2020].

[6] “Automotive Big Data Marketplace for Innovative Cross-sectorial Vehicle Data Services - Home,” https://automat-project.eu/, 2020, [Online; accessed 19-April-2020].

[7] Jan Zawadzki, “A Tale about a Giant, a Machine Learning pill, and the Automotive Industry,” https://towardsdatascience.com/a-tale-about-a-giant-a-machine-learning-pill-and-the-automotive-industry-1b339550dbad, January 2019, [Online; accessed 19-April-2020].

[8] J. Pillmann, B. Sliwa, J. Schmutzler, C. Ide, and C. Wietfeld, “Car-to-Cloud Communication Traffic Analysis Based on the Common Vehicle Information Model,” in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, June 2017, pp. 1–5.

[9] S. Herrnleben, M. Pfannemüller, C. Krupitzer, S. Kounev, M. Segata, F. Fastnacht, and M. Nigmann, “Towards Adaptive Car-to-Cloud Communication,” in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2019, pp. 119–124.

[10] J. Güldenring, L. Koring, P. Gorczak, and C. Wietfeld, “Heterogeneous Multilink Aggregation for Reliable UAV Communication in Maritime Search and Rescue Missions,” in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2019, pp. 215–220.

[11] Costin Raiciu and Christoph Paasch and Sebastien Barre and Alan Ford and Michio Honda and Fabien Duchene and Olivier Bonaventure and Mark Handley, “How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP,” in *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*.

San Jose, CA: USENIX Association, Apr. 2012, pp. 399–412. [Online]. Available: https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/raiciu

[12] B. Arzani, A. Gurney, S. Cheng, R. Guerin, and B. T. Loo, “Impact of Path Characteristics and Scheduling Policies on MPTCP Performance,” in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, May 2014, pp. 743–748.

[13] Chen, Yung-Chih and Lim, Yeon-sup and Gibbens, Richard J. and Nahum, Erich M. and Khalili, Ramin and Towsley, Don, “A Measurement-Based Study of MultiPath TCP Performance over Wireless Networks,” in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 455–468. [Online]. Available: https://doi.org/10.1145/2504730.2504751

[14] P. Hurtig, K. Grinnemo, A. Brunstrom, S. Ferlin, Ö. Alay, and N. Kuhn, “Low-Latency Scheduling in MPTCP,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 302–315, Feb 2019.

[15] R. Khalili, N. Gast, M. Popovic, and J. Le Boudec, “MPTCP Is Not Pareto-Optimal: Performance Issues and a Possible Solution,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1651–1665, 2013.

[16] Li, Feng and Chung, Jae and Jiang, Xiaoxiao and Claypool, Mark, *TCP CUBIC versus BBR on the Highway*, 03 2018, pp. 269–280.

[17] J. Pillmann, D. Behnke, B. Sliwa, M. Priebe, and C. Wietfeld, “Efficient and Reliable Car-to-Cloud Data Transfer Empowered by BBR-Enabled Network Coding,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.

[18] Wischik, Damon and Raiciu, Costin and Greenhalgh, Adam and Handley, Mark, “Design, Implementation and Evaluation of Congestion Control for Multipath TCP.” in *NSDI*, vol. 11, 2011, pp. 8–8.

[19] Ford, Alan and Raiciu, Costin and Handley, Mark and Bonaventure, Olivier and others, “TCP extensions for multipath operation with multiple addresses,” RFC 6824, Tech. Rep., 2013.

[20] Ford, Alan and Raiciu, Costin and Handley, Mark and Barre, Sebastien and Iyengar, Janardhan and others, “Architectural guidelines for multipath TCP development,” *IETF, Informational RFC*, vol. 6182, pp. 2070–1721, 2011.

[21] Raiciu, Costin and Handley, Mark and Wischik, Damon, “Coupled congestion control for multipath transport protocols,” IETF RFC 6356, Oct, Tech. Rep., 2011.

[22] Scharf, Michael and Ford, Alan, “Multipath TCP (MPTCP) application interface considerations,” RFC 6897, March, Tech. Rep., 2013.

[23] Huang, Junxian and Qian, Feng and Gerber, Alexandre and Mao, Z. Morley and Sen, Subhabrata and Spatscheck, Oliver, “A Close Examination of Performance and Power Characteristics of 4G LTE Networks,” in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 225–238. [Online]. Available: https://doi.org/10.1145/2307636.2307658

[24] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, “Mobile Data Offloading through Opportunistic Communications and Social Participation,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 821–834, 2012.

[25] Han, Bo and Qian, Feng and Hao, Shuai and Ji, Lusheng, "An Anatomy of Mobile Web Performance over Multipath TCP," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '15. New York, NY, USA: Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2716281.2836090

[26] Wikipedia, "Vehicle-to-everything," https://en.wikipedia.org/wiki/Vehicle-to-everything, 2020, [Online; accessed 19-April-2020].

[27] "3GPP A Global Initiative," https://www.3gpp.org/, 2020, [Online; accessed 10-May-2020].

[28] Chappell, Laura, "Inside the tcp handshake," *NetWare Connection*, 2000.

[29] Allman, Mark and Paxson, Vern and Stevens, William, "RFC2581: TCP congestion control," 1999.

[30] Mahdavi, Jamshid and Floyd, Sally, "TCP-friendly unicast rate-based flow control," *Note sent to end2end-interest mailing list*, vol. 28, 1997.

[31] Ha, Sangtae and Rhee, Injong and Xu, Lisong, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, p. 64–74, Jul. 2008. [Online]. Available: https://doi.org/10.1145/1400097.1400105

[32] D. Scholz, B. Jaeger, L. Schwaighofer, D. Raumer, F. Geyer, and G. Carle, "Towards a Deeper Understanding of TCP BBR Congestion Control," in *2018 IFIP Networking Conference (IFIP Networking) and Workshops*, 2018, pp. 1–9.

[33] Cao, Yi and Jain, Arpit and Sharma, Kriti and Balasubramanian, Aruna and Gandhi, Anshul, "When to Use and When Not to Use BBR: An Empirical Analysis and Evaluation Study," in *Proceedings of the Internet Measurement Conference*, ser. IMC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 130–136. [Online]. Available: https://doi.org/10.1145/3355369.3355579

[34] N. D. Adesh and A. Renuka, "Impact of traffic burst on the behavior of TCP variants for different LTE downlink schedulers," in *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2017, pp. 18–23.

[35] Dong, Pingping and Xie, Jingyun and Tang, Wensheng and Xiong, Naixue and Zhong, Hua and Vasilakos, Athanasios V, "Performance evaluation of multipath TCP scheduling algorithms," *IEEE Access*, vol. 7, pp. 29 818–29 825, 2019.

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

[37] B. Felix, I. Steuck, A. Santos, S. Secci, and M. Nogueira, "Redundant packet scheduling by uncorrelated paths in heterogeneous wireless networks," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00 498–00 503.

[38] F. Zhou, T. Dreibholz, X. Zhou, F. Fu, Y. Tan, and Q. Gan, "The Performance Impact of Buffer Sizes for Multi-path TCP in Internet Setups," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 2017, pp. 9–16.

[39] B. Lantz, N. Handigol, B. Heller, and V. Jeyakumar, "Introduction to Mininet," https://github.com/mininet/mininet/wiki/Introduction-to-Mininet, 2018, [Online; accessed 16-May-2020].

[40] "Socket, low-level networking interface," https://docs.python.org/3/library/socket.html, 2020, [Online; accessed 16-May-2020].

[41] Nathan Jennings, "Socket Programming in Python (Guide)," https://realpython.com/python-sockets/, [Online; accessed 20-April-2020].

[42] C. Paasch and S. Barre, "MultiPath TCP - Linux Kernel implementation," http://multipath-tcp.org/pmwiki.php/Users/ConfigureRouting, [Online; accessed 16-May-2020].

[43] "Wireshark," https://www.wireshark.org/, [Online; accessed 3-July-2020].

[44] Volker Gropp, "bwm-ng(1) - Linux man page," https://linux.die.net/man/1/bwm-ng, [Online; accessed 26-March-2020].

[45] "Mininet Python API Reference Manual," http://mininet.org/api/classmininet_1_1link_1_1Intf.html#, 2018, [Online; accessed 16-May-2020].

[46] Johannes Güldenring, "rural_kohlfurt and highway_A535," https://www.youtube.com/watch?v=RfVmz5xJWas&feature=youtu.be, 14 July 2020, [Online; accessed 05-August-2020].

[47] "3G / 4G / 5G coverage map, Germany," https://www.nperf.com/en/map/DE, 2020, [Online; accessed 05-August-2020].

[48] Chris Lim, "Buying a SIM Card in Germany: Telekom vs Vodafone vs O2," https://nomadtoolkit.com/buying-sim-card/germany-telekom-vodafone-o2/, 2020, [Online; accessed 04-July-2020].

List of Figures

List of Tables