Journal of Network and Systems Management (2023) 31:49
https://doi.org/10.1007/s10922-023-09737-0

Performance Analysis of the Multi-connection Tactile Internet Protocol over 5G

Delia Rico[1] · Karl-Johan Grinemmo[2] · Anna Brunstrom[2] · Pedro Merino[1]

Received: 29 November 2022 / Revised: 23 March 2023 / Accepted: 14 April 2023 /
Published online: 1 June 2023

Abstract

Tactile Internet is an Internet network that combines ultra-low latency with extremely high availability and reliability. Since traditional protocols, such as UDP and TCP, cannot support this operation, other transport protocols are required to meet the stringent requirements of the Tactile Internet. This paper evaluates the implementation of the Multi-connection Tactile Internet Protocol (MTIP), a multi-connectivity transport protocol for the Tactile Internet. MTIP uses application and network status information to select network paths intelligently and, in so doing, to improve reliability and latency. The paper studies how different configurations of the MTIP algorithm impact its path selection and the effect on lost and late packets. This evaluation is performed in an emulated environment and in a 4G/5G lab to evaluate the protocol in diverse scenarios. The results show a direct trade-off between higher reliability requirements and the number of duplicate packets.

Keywords Multi-connectivity · Remote control · Transport protocol · Tactile Internet · Industrial robot · Context awareness

✉ Delia Rico
deliarico@uma.es

Karl-Johan Grinemmo
karl-johan.grinnemo@kau.se

Anna Brunstrom
anna.brunstrom@kau.se

Pedro Merino
pmerino@uma.es

[1] ITIS Software, Universidad de Malaga, Malaga, Spain

[2] Computer Science, Karlstad University, Karlstad, Sweden

1 Introduction

Tactile Internet (TI) requires low levels of latency combined with high levels of reliability [1], something which makes traditional transport protocols, such as UDP [2] and TCP [3], less than ideal. Instead, real-time transport protocols have traditionally been used in these networks, e.g., IRTP [4], ETP [5], HMTP [6], STRON [7], and Smoothed SCTP [8]. However, in recent years numerous end-to-end solutions have been developed for wireless and cellular networks, such as network awareness, multi-homed devices, and private network slices, among others [9]; these real-time transport protocols are not suited to take advantage of these current advancements.

Due to this evolution, a growing interest has been in researching novel protocols or extensions more aligned with current trends [10]. Examples of these new protocols and extensions include MPTCP [11], QUIC [12], MPQUIC [13], and MPRTP [14]. However, even if these protocols follow current trends, they are not directly aimed at complying with the requirements of a Tactile Internet. Not least, they lack the flexibility to adapt to the specific requirements of the different Tactile Internet applications [1]. Moreover, they come with extra features not needed for Tactile Internet, which could worsen their performance in these networks due to the added complexity and unnecessary header overhead.

Remote operation in a Tactile Internet requires a lightweight protocol flexible enough to adapt to specific preferences and unexpected network conditions. A transport protocol for remote operations should optimally trade latency and reliability to accommodate application requirements. To this end, we propose the Multi-connection Tactile Internet Protocol (MTIP), a multipath protocol specifically designed for the remote control of a Tactile Internet in current and next-generation wireless networks. MTIP manages available network paths according to application preferences to offer a tunable quality to applications requiring specific latency and high-reliability levels.

This paper presents a proof-of-concept implementation and evaluation of MTIP [15]. It extends the work presented in [16], introducing an evaluation of MTIP in an emulated environment and an additional evaluation of the protocol using an industrial robot in a real 4G/5G testbed located at the University of Malaga [17]. The aim is to assess MTIP's ability to intelligently select network paths in diverse scenarios and show how MTIP can be configured to offer reliable, low-latency transport services at the cost of sending some redundant packets in an actual application.

The remainder of the paper is organized as follows. Section 2 surveys related work. Section 3 discusses the design of MTIP and Sect. 4 its implementation. Section 5 presents a first evaluation of MTIP's ability to select network paths and meet the requirements of Tactile Internet applications in an emulated environment, and Sect. 6 shows a proof-of-concept evaluation of remotely controlling an industrial robot in a real 4G/5G testbed, in order to validate our previous results and illustrate the operation of MTIP in a real scenario. The paper concludes in Sect. 7.

2 Related Work

Real-time multimedia protocols have frequently been used to support Tactile Internet-like applications, e.g., RTP [18] and SCTP [19], two commonly used real-time multimedia protocols, are often used for remote control applications [20, 21]. In addition to RTP and SCTP, several specific real-time protocols and extensions have been adapted for haptic control and robotic teleoperations: Smoothed SCTP [22] is an extension of SCTP, which separates different types of traffic to reduce jitter; the Interactive Real-Time Protocol (IRTP) [23] is a protocol that combines connection-oriented and unreliable data to improve performance; and the Real Time Network Protocol (RTNP) [24] is a protocol that reduces internal delays in a specific multitasking operating system. Still, Tactile Internet remote control data possess characteristics distinct from real-time multimedia streams. In particular, remote control packets are usually smaller; they are transmitted at lower data rates, and real-time multimedia protocols operations like flow control could impose extra delay and variations on these packets, severely impeding Tactile Internet control operations.

A more comprehensive survey of Tactile Internet protocols can be found in Kokkonis et al. [25], and Antonakoglou et al. [26], with additional examples of protocols aligned with TI communications such as the Efficient Transport Protocol (ETP) [5], a protocol that adapts the inter-packet gap to network congestion conditions to reduce the round trip time, STRON [7], a protocol that uses forward error correction to reduce the latency of media streams, and HMTP [6], a multicast protocol for synchronous collaboration in a haptic network. While they collect promising solutions, these do not fully align with the potential of novel networks. Current networks offer opportunities such as information exposure and the ability to utilize multiple paths, which is a key trend in novel protocols (further information on novel trends in [9, 10]).

Multi-connectivity is the ability to manage multiple connection paths with a single protocol and opens up for more reliable data transmission without compromising latency. Multi-connectivity has grown in importance lately with the expansion of multi-homed devices and multiple access technologies in 5G networks [27]. Some of the most well-known protocols that support multi-connectivity are Multipath TCP (MPTCP) [11], Multipath QUIC (MPQUIC) [13, 28], and Multipath RTP (MPRTP) [14]. However, these protocols are not aimed at Tactile Internet communications, (e.g., [29–32]). Mainly, they lack the adaptability necessary to adapt to the varying demands of different Tactile Internet applications [1]. Furthermore, they contain a number of unnecessary features for the Tactile Internet, which could cost latency without any benefit. Tactile Internet applications need a protocol that can adapt to the specific requirements of each use case and which can take advantage of the full potential of the underlying networks. Compared with all this previous proposals, we designed the Multi-connection Tactile Internet Protocol (MTIP), a protocol that exploits multi-connectivity and context awareness to achieve the lowest latency and high (but not full) reliability with the available communication technologies.

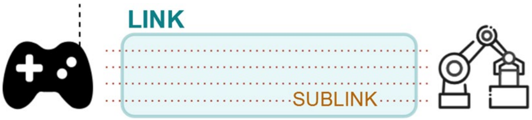
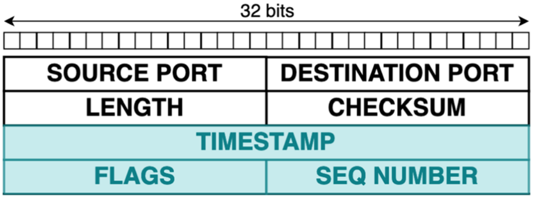


Fig. 1 Use of multiple sublinks in MTIP



SOURCE PORT	DESTINATION PORT
LENGTH	CHECKSUM
TIMESTAMP	
FLAGS	SEQ NUMBER

Fig. 2 The MTIP header

3 Description of the Protocol

MTIP is a transport protocol for the remote control of Tactile Internet applications. MTIP exploits the benefits of having several available paths by intelligently sending redundant packets over multiple paths or sublinks, which act as a reliable link (see Fig. 1).

In order to provide adequate transport services to tactile applications with the least redundant network resources, MTIP uses context awareness to manage and schedule traffic over multiple paths flexibly. The protocol considers both application preferences through its Application Programming Interface (API) and information about the network status through collecting network information to select the best sublinks to send data.

In the remainder of this section, we provide an overview of the MTIP protocol, emphasizing its sublink management and how it uses knowledge about application requirements and network conditions to decide which sublinks to transfer data packets.

3.1 The MTIP Packet

The structure of an MTIP packet is illustrated in Fig. 2. MTIP packets use a UDP-like header with a few additional fields. The timestamp is used to control the deadline of the packets; the sequence number controls the order and identifies duplicate packets, while the flags differentiate the two types of packet used on MTIP, i.e., data and control packets.

Data packets carry application data in the payload. These packets can be duplicated over several sublinks depending on the application preferences. On the receiving side, they are processed to filter out late or duplicate packets, confirmed by *ACK* messages, informing about the reception state. In contrast, control packets are not filtered out but are directly processed and acknowledged. They manage link establishment and additional operations. Control packets of the types *Link* and *Preference* are sent to establish the link, share information about the available interfaces on each endpoint, and synchronize application preferences. Other control packets include *Keepalive*, employed to perform periodic measurements on idle sublinks, and *Characterization*, used when the application demands more detailed measurements. MTIP uses *Finish* packets to close a communication session.

3.2 Data Transmission

MTIP can send data using all paths or only a subset of the available paths. This decision can be left to the application, although MTIP includes an internal algorithm, i.e., the MTIP sending algorithm, to make this selection. While using all paths to send data duplicated simultaneously could be the most beneficial by maximizing some Key Performance Indicators (KPI), such as reliability, in most cases, it would also waste unnecessary network resources. The MTIP sending algorithm can adapt to conditions by maximizing the KPI values and minimizing the waste of resources. As shown in Fig. 3, it makes two decisions: (i) the number of sublinks to use and (ii) which sublinks to use.

Initially, MTIP starts sending data packets on all sublinks. If the receiving endpoints get too many duplicate packets, MTIP reduces the number of sublinks it uses. In contrast, if reducing the number of sublinks results in too many packets arriving late or getting lost, MTIP increments the number of sublinks in use. This decision is based on the application preferences *Duplicate Threshold* and *Loss-Late Threshold*, which indicate the percentage of duplicate and late or lost packets considered acceptable. Then, MTIP decides which sublinks to use for the transmission by ranking available sublinks in terms of latency and reliability. The ranking is based on the application preference *Latency Weight*, which reflects the relative significance of the application's latency versus reliability. A higher latency weight would rank the sublinks taking the latency KPI more into account, while a lower one would rank them prioritizing the reliability KPI.

On the receiving side, data packets go through the MTIP reception algorithm to determine if they should be sent to the application layer or discarded before being acknowledged. A representation of the MTIP receiving algorithm is shown in Fig. 4. In particular, MTIP keeps a window to control the arrival of packets: packets that are not deemed obsolete enter the reception window. MTIP uses a timestamp in the packet header to control time. MTIP assumes that both endpoints are synchronized,

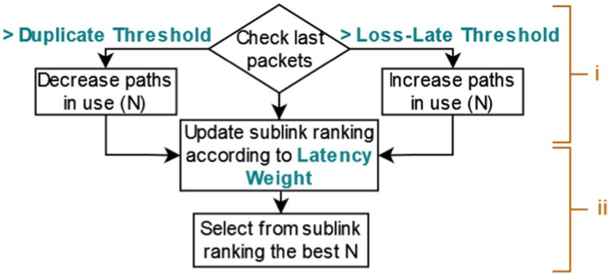

> Duplicate Threshold
Check last packets
> Loss-Late Threshold
Decrease paths in use (N)
Increase paths in use (N)
Update sublink ranking according to Latency Weight
Select from sublink ranking the best N
i
ii

Fig. 3 MTIP sending algorithm

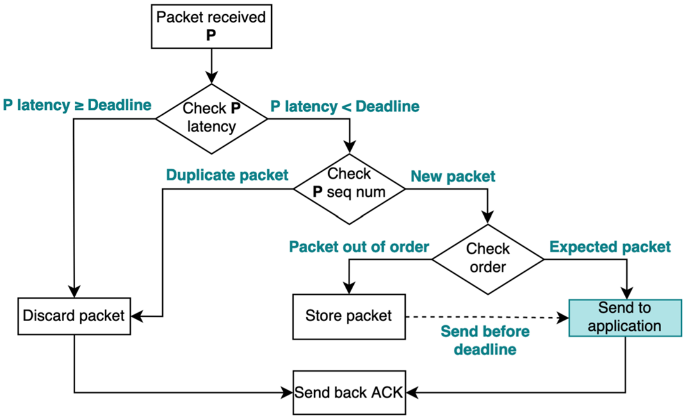


Fig. 4 MTIP receiving algorithm

e.g., by using the Precision Time Protocol (PTP) [33]. MTIP features a mechanism that prevents duplicate packets from being delivered to upper layers: duplicate packets share the same sequence number and are filtered out by MTIP. When a packet arrives on the receiving side, its sequence number is checked against a list containing the sequence numbers of packets already received. MTIP also prevents the delivery of out-of-order messages. If a packet arrives out of order, MTIP stores it until i) the missing packet arrives, and they are both sent to the application layer in order, or ii) the missing packet is considered lost, and just the initial packet is sent to the application before it reaches its deadline. The deadline of a packet is calculated using the timestamps and the application preferences on the maximum one-way latency supported. A packet is considered lost when its deadline has passed (refer to [34] for further information about multipath packet expiration).

3.3 Context Awareness

To offer an enhanced service, MTIP uses information outside the transport layer's scope. This information can be divided into network awareness and application awareness.

Through keepalive and characterization packets, MTIP collects network state information which it stores as network KPIs for each sublink (downlink and uplink). This information is synchronized on both endpoints through ACK packets. The most relevant KPIs are latency and reliability. Latency is calculated using the timestamp on the packets and the current time, while reliability is calculated by measuring packet loss through sequence numbers.

MTIP exposes an API similar to the Socket API. In contrast to the Socket API, it includes functions to manage network and application awareness. These functions are listed in Table 1. Application preferences can be separated into communication preferences and protocol configurations. Communication preferences inform MTIP about application transport-service requirements, e.g., the maximum acceptable

Table 1 Extension of the Socket API for MTIP

Function	Description
socket()	Create a MTIP socket
bind()	Bind interfaces to the socket
link()	Establish a MTIP link
send()	Send data on a selected link
receive()	Receive data on a selected link
set_preferences()	Modify preferences and protocol configurations
get_feedback()	Retrieve information about network status

one-way latency and the parameters duplicate threshold, loss-late threshold, and latency weight (cf. Figs. 3 and 4). Protocol configurations have default values and define how MTIP works internally, such as the interval between keepalive packets, the importance of new measurements over older ones, and the number of relevant packets used for the measurements and the thresholds. The application can also define these protocol configurations.

4 Implementation

We have developed the first implementation of the MTIP protocol in C++ using the open-source Qt library [35], benefiting from Qt's interprocess communication and networking modules. The implementation follows the state diagram illustrated in Fig. 5, which shows the different states of an MTIP link during its lifetime. An MTIP link begins in the CLOSED state, and once it has been established, it enters the LINKED state, where the data exchange can start. At the end of a transport session, the MTIP link is closed, which is done similarly to TCP, i.e., via a FIN_WAIT state.

The implementation of MTIP is depicted in Fig. 6. It exposes an API compatible with C that developers can use to create applications. When a link has

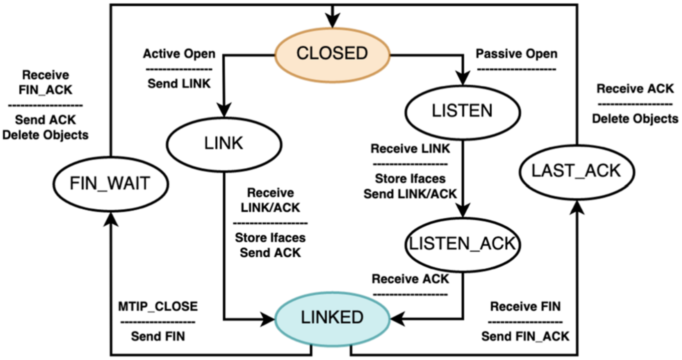


Fig. 5 MTIP state machine

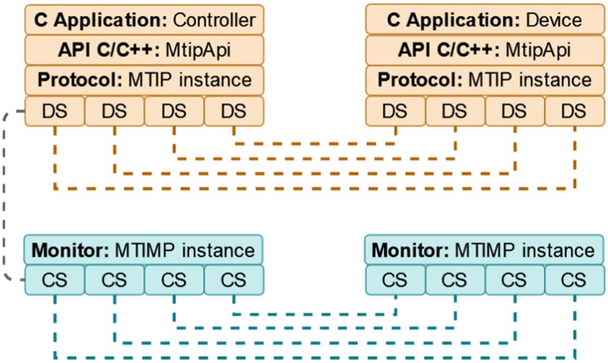


Fig. 6 MTIP implementation structure

been established and both endpoints have synchronized their information about the sublinks, MTIP creates sockets for each sublink. These sockets are separated into Data Sublinks (DS) and Control Sublinks (CS) but are controlled by the protocol as a single link. The DS are used to send the application data following the algorithms presented in Sect. 3.2 with the application preferences presented in Sect. 3.3. The CS, on the other hand, are in charge of sending control packets to take measurements of the network and keep network KPIs up to date. The operation of CS is supervised by the control plane or monitoring entity (MTIMP), which is in charge of storing the characterization information used by the protocol. The information is stored and shared to avoid unnecessary copies of data, using callback references and signals to keep the operation lightweight.

5 Evaluation in an Emulated Environment

As a first step, we study the impact of different configurations of the MTIP algorithm in an emulated test setup. The scenarios considered have been selected to evaluate the extent to which MTIP can deliver a requested service when it experiences packet loss and packet delays. The topology under test comprises two endpoints connected by four independent sublinks, as shown in Fig. 1, where the remote controller sends messages at the typical Tactile Internet rate of 1 kHz [1].

The evaluation platform consists of a Ubuntu computer with 16 CPUs and 64 GB RAM. We use the Mininet tool [36] to create a virtual topology with independent network paths. We apply `netem` rules with the `tc` tool [37] to emulate different network impairments, including random packet losses and delays. Each test is executed 10 times to have accurate measurements. The results presented in Sect. 5.3 show the mean values of the executions. We omit all iterations from the figures, since the 95% confidence interval lower and upper bounds rarely exceed ± 5% and, in any case, never affect the tendency of the effect produced by the thresholds under study.

5.1 Scenarios

We have designed three scenarios that differ in the choice of KPI being impaired:

- *Delay scenarios* emulate delays in the lower layers of the protocol stack. These delays could be caused by different configurations, such as the acknowledged mode of the Radio Link Control protocol (RLC) [38], which provides higher reliability at the cost of longer delays. Delay scenarios explore cases with significant delay variation and no losses.
- *Loss scenarios* emulate lower-layer losses, e.g., losses caused by the unacknowledged mode of RLC. Loss scenarios explore cases with losses but with no significant delay variations.
- *Mixed scenarios* are the most realistic of the three types of scenarios. They present the case of having both delay variation and losses caused by balanced configurations of lower layers.

All scenarios change their conditions every second. However, we create three variants of each scenario: one with good conditions on all paths, one with dynamic conditions on all paths, i.e., paths that change from good to poor conditions and vice versa, and one with poor conditions on all paths, in order to showcase a extremely bad situation. To have some reference values on the worst and best-case scenarios, we characterize the scenarios using MTIP over the paths separately, showing the average value, and then using MTIP redundantly on all paths simultaneously. We show the results of this characterization in Table 2. It is important to note that the percentage shown in the table is the amount of late and lost packets that do not reach the application layer. In delay scenarios, the percentage represents the fraction of packets whose transfer time is larger than 10 ms, a typical deadline in Tactile Internet [1]; in loss scenarios, the percentage denotes actual packet losses; while in mixed scenarios, the percentage is a combination of packets that are late and lost.

5.2 Measurements

To evaluate the operation of the MTIP algorithm, we study the impact of its primary parameters: the loss-late threshold, the duplicate threshold, and the latency

Table 2 Characterization of the testing scenarios

	Delay scenarios			Loss scenarios			Mixed scenarios		
	Good (%)	Dynamic (%)	Poor (%)	Good (%)	Dynamic (%)	Poor (%)	Good (%)	Dynamic (%)	Poor (%)
One path	0.12	9.72	16.30	0.86	3.89	10.52	1.92	11.58	25.02
All paths	0.01	0.02	0.02	0.00	0.01	0.02	0.07	0.06	0.89

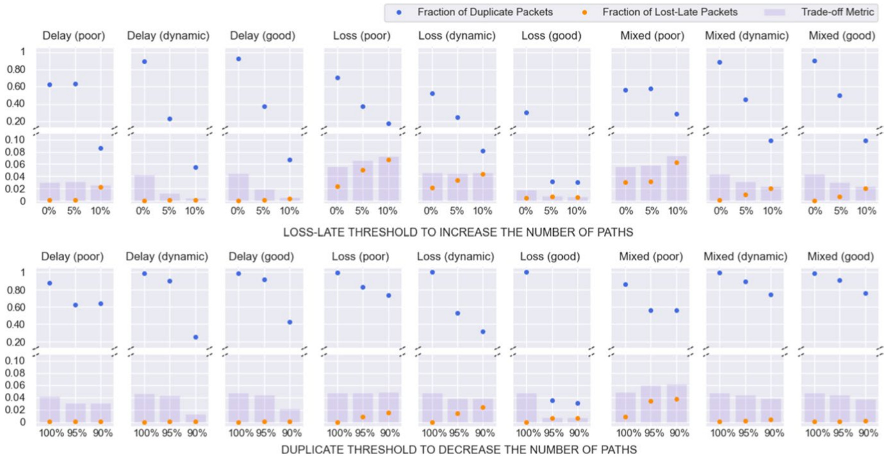


Fig. 7 Effect of the loss-late threshold (above) and the duplicate threshold (below) on the different scenarios

weight (cf. Fig. 3). We measure the trade-off between the reliability achieved and the number of wasted network resources. The reliability is measured by counting the amount of lost or late packets. In contrast, the waste of network resources is measured in terms of the number of unnecessary duplicate packets that arrive at the receiving endpoint. In order to showcase this trade-off in the graphs, we have created the *trade-off* metric. This metric directly compares duplicates and losses and is introduced to make it easier to compare the outcome from different scenarios. Since the rate of duplicate packets is usually higher and less critical than losses, the weight is set to reflect that.

$$Trade\text{-}off = \frac{Losses + (Duplicates \times Weight)}{1 + Weight}$$

Our figures show the values of the *trade-off* metric with $Weight = 0.05$. This value represents the importance of wasting resources, shown by the number of duplicate packets, set to a 5% compared to the importance of receiving correct messages. This value helps us compare lost, late, and duplicate packets, see where the metric is lower, and determine which case is more beneficial. However, the decision on the actual value of the metric depends on the specific target application and the importance it assigns to reliability versus resource usage.

5.3 Results and Discussion

First, we study the effects of varying the loss-late and duplicate thresholds. In the topmost graphs of Fig. 7, the loss-late threshold is varied while the duplicate threshold is fixed, and in the bottommost graphs, the duplicate threshold is varied while the loss-late threshold is kept fixed. The fixed thresholds have a value of 5%, enabling

MTIP to adapt to changes in the network without being too restrictive, allowing us to study the effect of the varied threshold.

The graphs suggest that a lower loss-late or a higher duplicate threshold causes a higher number of duplicate packets and fewer losses. In these cases, MTIP is more likely to increase than decrease the number of sublinks. The result is the opposite if the loss-late threshold is set to a higher value or the duplicate threshold is set to a lower value, with fewer network resources used at the cost of some packet losses. This is seen more clearly in worse scenarios, like the Loss (poor) one, since in fairly good scenarios, the variation on the lost-late packets is minimum.

The trade-off metric allows us to analyze the trade-off between lost-late and duplicate packets. In general, if the scenario is good, it benefits from reducing duplicates; while in worse scenarios, it depends on the specific case. For instance, in the mixed scenario (poor), both 0% and 5% loss-late threshold configurations are considered better than the 10% configuration. The reason is that with a 10% configuration, lost-late packets increase more than desirable, even if duplicates are reduced.

The last parameter under study is the latency weight used to rank and select the sublinks. This parameter can be evaluated in mixed scenarios where delays and losses could be used to rank the sublinks. In Fig. 8, we set the values of the loss-late and duplicate thresholds to 10% and 90%, respectively, to have scenarios with some losses, and evaluate the effect of the latency weight parameter on lost-late and duplicate packets. The results show that in better scenarios, a selection of paths that takes into account latency measurements (i.e., latency weight higher than 0%) is generally more beneficial than a selection that only considers reliability (i.e., latency weight 0%). This is because reliability is already improved by using multiple paths; so a selection considering latency measurements would also include this KPI in the equation, having richer information on which to perform the selection.

Furthermore, we can see that there is less variation of lost-late packets than there is a variation of duplicate packets. Duplicate packets are usually more affected by an inappropriate ranking. If the sublinks are stable and the demands are not met with the sublinks currently in use, the algorithm decides to use more paths, which causes more duplicates. However, it does not affects lost-late packets that much after stabilizing. In Fig. 9, we can see why this effect significantly impacts better scenarios.

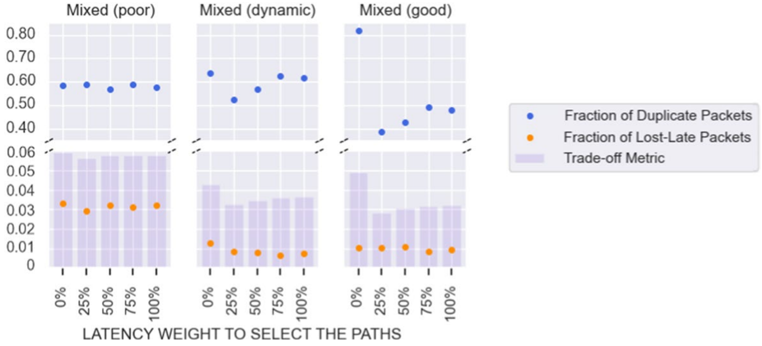
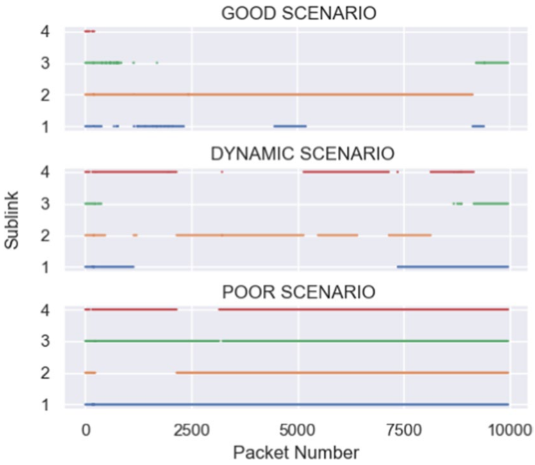


Fig. 8 Effect of the latency weight on lost-late and duplicate packets

Fig. 9 Sublinks selected in the different scenarios



We show an example of the typical sublinks selected in the different scenarios. In an exemplary scenario, a proper selection usually uses only one sublink; in a dynamic scenario, the selection includes more sublinks; and in a poor scenario, almost all sublinks are used at all times. Therefore, in this latter case, the specific ranking of the sublinks is insignificant if all of them are in use.

6 Evaluation in a Real Environment

After studying the performance of MTIP in an emulated environment, we consider a more realistic evaluation in a 4G/5G testbed. The objective of our testbed evaluation is to check the validity of our emulated tests, i.e., to confirm that the results obtained in the emulation evaluation of MTIP apply to a real-case scenario, and, secondly, to demonstrate that MTIP could be used in an actual use case. For our tests, we remotely control an industrial robot using MTIP on the Morse lab [17], located at the University of Malaga (UMA). This lab offers an infrastructure that deploys a fully private network in collaboration with Telefónica. The network supports LTE. It also supports 5G NR, enabling Non-Stand Alone and Stand Alone 5G tests. In the following subsection, we provide more details of the topology and the application.

6.1 Topology and Use Case

The experimental setup in our 4G/5G testbed is shown in Fig. 10. The setup consists of a computer that acts as a remote controller and a second computer that acts as an onboard device of an Unmanned Ground Vehicle (UGV). The computers are connected using multiple paths, and are also connected to a PTP server for synchronization at a nanosecond level (the PTP server is not shown to avoid cluttering the figure).

The computers run the Ubuntu operating system with a low-latency kernel [39]. Four different networks connect them: a 4G network, a 5G NSA network with

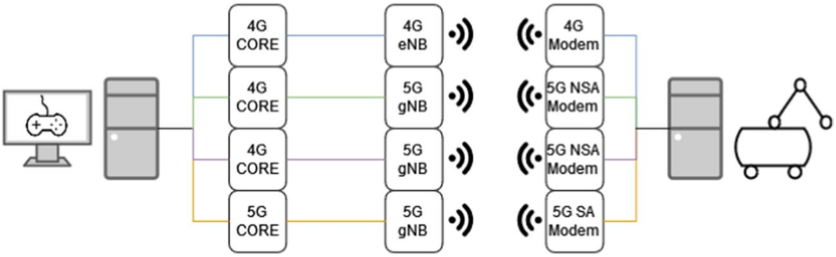
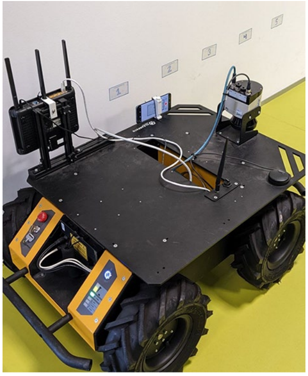


Fig. 10 Experimental setup overview

special QCI (QoS Class identifier) priority, a secondary regular 5G NSA network, and a 5G SA network. The robot used in the tests is the Husky Clearpath robot [40] shown in Fig. 11. The Husky robot is a medium-sized UGV that runs the Robot Operating System (ROS), an open-source set of software libraries and tools created to build robot applications [41]. In order to run ROS applications over MTIP, we create a proxy setup that encapsulates ROS messages over MTIP. Figure 12 presents the resulting communication setup.

The case study we want to replicate in our tests is that of a Tactile Internet remote control of an industrial robot. In this application, the industrial robot must move within the specific deadline in a coordinated manner with other robots on the shop floor, as in an assembly line. It must also work reliably; however, it is a priority to do so within the deadline or not move at all. Otherwise, it could cause an accident or destroy important devices. Thus, when a deadline is missed, the robot does not move and misses one of its actions at the assembly line. The application should use resources wisely since it could affect both the network and the devices. Resources can be wasted both as a result of the robot missing actions at the assembly line and as a result of redundant communication. In the case study context, our trade-off metric with a weight of 0.05 captures a scenario where a missed action at the assembly line is twenty times more wasteful than sending a redundant message, representing a more efficient overall operation with a lower trade-off metric. In terms of data characteristics, this application is analogous to the one presented in the emulated

Fig. 11 The Husky Clearpath robot used in the testbed tests



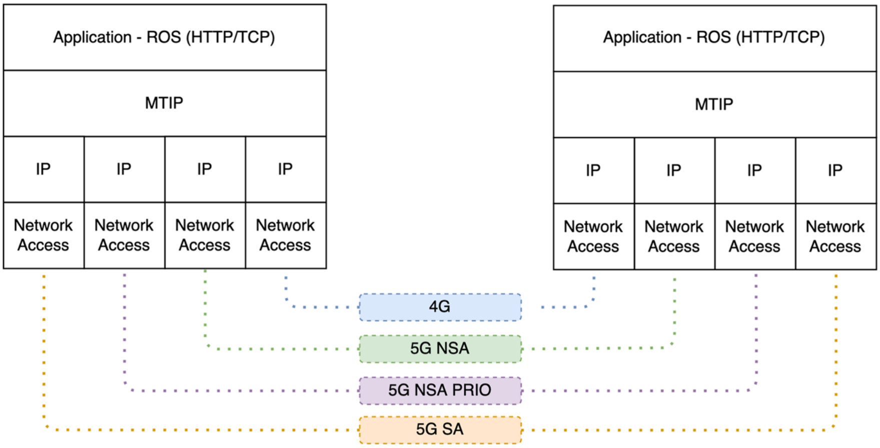


Fig. 12 Communication setup between the remote controller and the robot

evaluation, i.e., sending an order to move the robot at a constant rate with a 10 ms deadline. To allow the robot in use to process the commands and the movement to be perceptible, the rate has been reduced to 0.2kHz, a value still within reason for Tactile Internet applications [1].

6.2 Scenarios

To select the scenarios to evaluate MTIP in a real environment, we follow a similar approach as the one presented in the emulated environment. However, we have to keep in mind the characteristics of a real network and its variability in its normal state. For this reason, all the scenarios presented in this environment are dynamic and can be compared with the dynamic variants present in the previous evaluation.

The first scenario presented is the *regular scenario*. This scenario introduces the normal operation of the network without any additional impairments. In the network under test and any real network, there is always some end-to-end delay due to the physical distance and limitations of the devices. This delay is variable and cannot be controlled by the end user. In terms of losses, we are working with a stable network that has almost no losses when using Tactile Internet data since packets are fairly small and are sent at a relatively low rate. Therefore, the first scenario presented in this environment is comparable to the previous *Delay scenarios* of the emulated evaluation.

The second scenario under study is the *impaired scenario* and is comparable to the *Mixed scenarios*. Since the variability on the end-to-end delay is not something we can remove, adding losses to the scenario will make the scenario seem closer to the previous *Mixed scenario* than the previous *Loss scenario*. This scenario will consist of the delay variation created by the network and some loss impairment that will be added to all sublinks in the communication. In order to add these losses to the network, we perform some real measurements of the losses caused in the 4G

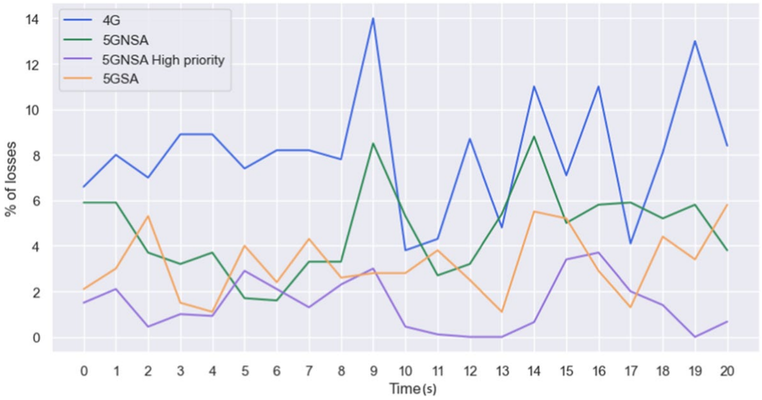


Fig. 13 Losses caused with the anechoic chamber

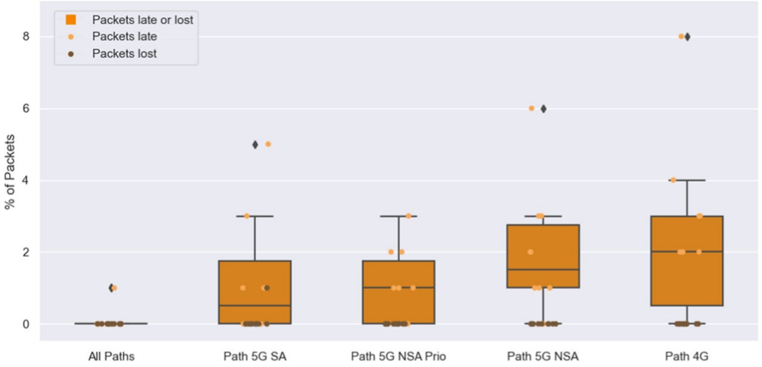


Fig. 14 Characterization of the regular scenario

and 5G modems when using a multi-Probe Anechoic Chamber to create more challenging radio conditions. An example of the real values of the losses captured can be seen in Fig. 13. Nevertheless, physically replicating this exact behavior multiple times is nearly impossible, since losses keep varying on each path through time. Thus, in order to have reproducibility, we add these real values of loss impairments to the network using the network emulation tool `netem` and the `tc` tool [37].

We characterize the scenarios to show the impact on the losses and the number of packets that usually arrive after the desired 10 ms deadline. The results of the characterization of the scenarios are presented in Figs. 14 and 15. All tests presented in this section show both the boxplot and the individual results of each of the 10 iterations performed in order to analyze with more detail the results in this dynamic scenario.

In the characterization of the regular scenario (i.e., Fig. 14), we can see how the 5G SA and the 5G NSA PRIO usually work better than the rest. The 5G NSA PRIO is a 5G NSA connection with a special priority over other data that is going through

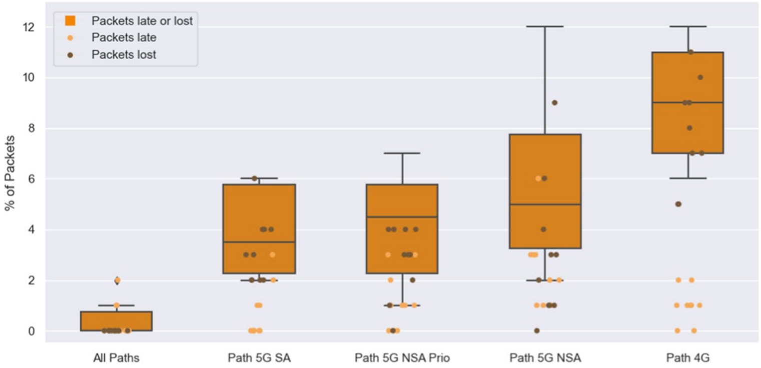


Fig. 15 Characterization of the impaired scenario

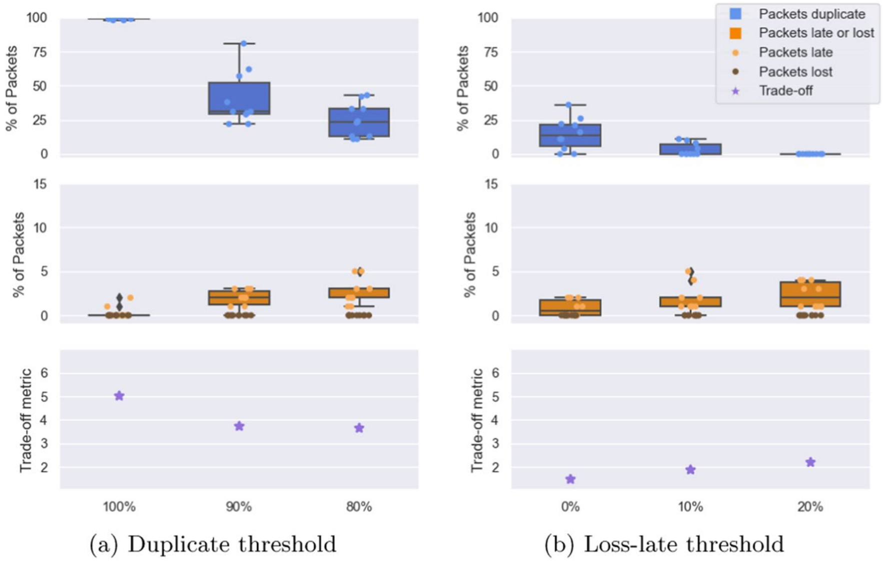
the network, so it is less affected by delay and losses. Next, we have the regular 5G NSA and the 4G paths. In general, this regular scenario presents a fairly good network that has almost no losses and usually a small amount of packets arriving later than 10 ms (medians lower than 2%). The left boxplot shows the result of using all paths at the same time, resulting in almost no late packets.

In the characterization of the impaired scenario (i.e., Fig. 15), we add the losses presented in the previous Fig. 13. The loss impairments act differently depending on the network but still show a trend similar to the one seen in the previous characterization. This trend means that 5GNSA PRIO and 5G SA have fewer losses, the 5G NSA has more impairments, and the 4G network has the most losses. We can see the measured impact in Fig. 15, resulting in a scenario similar to the regular one in terms of late packets, but with a clear difference concerning the impact of the losses. Nevertheless, when using all paths, we can see a situation similar to the previous one, i.e., a scenario with very few late and lost packets.

6.3 Measurements

We evaluate the performance of MTIP at the transport and application layers. The measurements conducted at the transport layer are comparable to the ones conducted in the emulated environment. We measure the number of duplicates and of late and lost packets when increasing the loss-late threshold, decreasing the duplicate threshold, and evaluating different settings of the latency weight. Moreover, we use the trade-off metric, as presented in Sect. 5.2, to show the most beneficial cases for the environment under study. The graphs presented in Sects. 6.4.1 and 6.4.2 show the percentage of duplicates, the percentage of late-lost packets, and the aforementioned trade-off metric.

At the application layer, we measure the accuracy in terms of the percentage of robot actions that reach the robot and, thus the number of movements correctly processed by it. We also tie these results to the trade-off metric, which can help us assess efficiency, since lower values of the metric show less waste of resources for the target application. The combination of both measurements allows for a more



(a) Duplicate threshold (b) Loss-late threshold

Fig. 16 Impact of the thresholds in the regular scenario

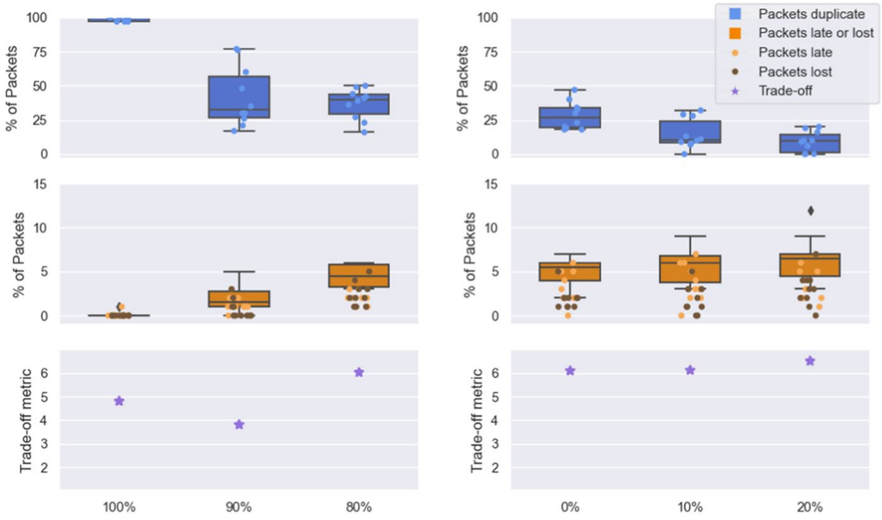
comprehensive analysis of the most favorable cases presented. In the following subsection, we provide the results of the evaluation.

6.4 Results and Discussion

6.4.1 Duplicate and Loss-Late Thresholds

First, we study the impact of the thresholds in the regular scenario. As in the previous evaluation, we assess the impact of each threshold while setting the others to a value of 5%, a value selected to see variations and assess the effect of the threshold that is being modified. Figure 16a and b shows the number of duplicates and lost or late packets when decreasing the duplicate threshold and increasing the loss-late threshold.

The behavior is analogous to the one presented in the emulated scenario, with an increase in the duplicate packets tied to the decrease of late packets due to the effect of redundancy; and vice versa. Focusing on the trade-off metric when modifying the duplicate threshold, we see that reducing the number of duplicate packets enhances the trade-off as in the 80% duplicate threshold. Since the regular scenario is not highly impaired, reducing the duplicates does not significantly affect the number of late packets. However, focusing on the loss-late threshold variation, we see that having some redundancy has a beneficial effect on the trade-off as in the 0% loss-late threshold. Since the scenario is not ideal, it can benefit from some redundancy. Specifically, both evaluations show a more advantageous case when keeping duplicate packets around 20–25%.



(a) Duplicate threshold (b) Loss-late threshold

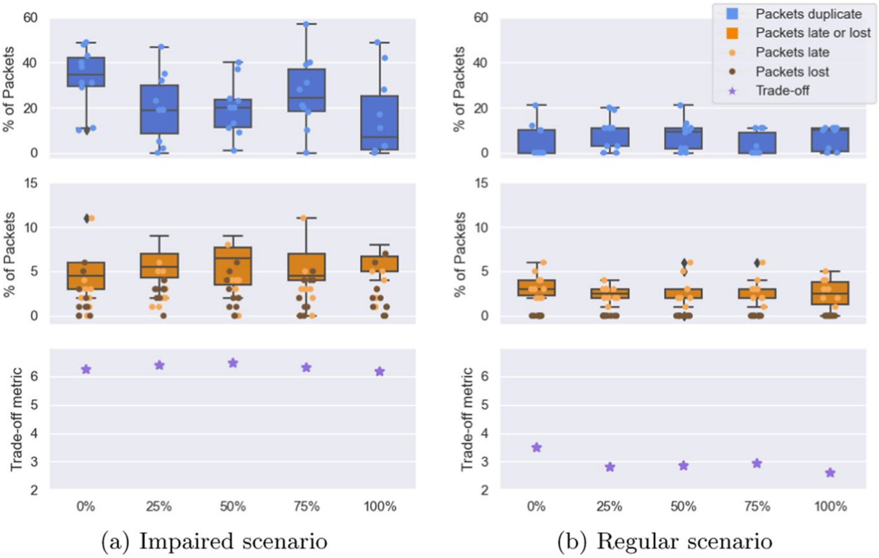
Fig. 17 Impact of the thresholds in the impaired scenario

In Fig. 17a and b, we show the results of the impaired scenario. The first impression is the same as in the regular scenario; lower values of the loss-late threshold or higher values of the duplicate threshold cause more duplicates and fewer losses. However, considering the trade-off metric, we can see that reducing the duplicate packets too much in both the duplicate threshold case or the loss-late threshold case results in a higher and less desirable trade-off metric value (see the 80% duplicate threshold or the 20% loss-late threshold). Increasing the number of duplicate packets in the last case, i.e., having around 50% of duplicate packets, seems more beneficial in this impaired case since the scenario is more prone to losses and can benefit from a higher redundancy. Nevertheless, this does not mean that a 100% of duplicate packets would be the optimal case, since the trade-off metric shows that further increasing the duplicate packets does not significantly improve the losses and this could be considered a waste of resources in the scenario under study.

6.4.2 Latency Weight

The result of the evaluation of the latency weight is shown in Fig. 18. As in the previous evaluation, we set the values of the thresholds to study the impact from only using latency measurements (100%) to only using reliability measurements (0%) to select the paths from which to send data. We study both scenarios presented since both can have losses, even if this is not likely in the regular scenario.

The results show that, in general, the difference in the configuration of the latency weight does not significantly affect the trade-off metric. In the impaired scenario, we see variations in duplicate and lost-late packets. Specifically, we see that taking into account latency measurements (25%–100%) can reduce the number of duplicates.



(a) Impaired scenario

(b) Regular scenario

Fig. 18 Impact of the latency weight in the scenarios

However, with the current configuration of the trade-off metric, this reduction of duplicates does not significantly affect the metric; since it incurs in few, but some losses that also impact the metric. Moreover, as in the emulated environment, this poor scenario usually makes use of several paths at the same time, so that the specific selection of paths is not as important as the number of paths in use. In the regular scenario, the difference is slightly more noticeable in the trade-off metric. This shows that using just reliability measurements (0%) can be less convenient than using latency measurements as well. However, since the scenario is dynamic and the losses and late packets are not numerous and can occur in any of the paths, once a path is generally stable, the ranking is no longer as critical.

The behaviors mentioned can be better understood with some examples of the sublinks selected to send data in each scenario, as shown in Figs. 19 and 20. In the impaired scenario, we see a tendency to use more paths. In contrast, in the regular scenario, we can see how the protocol adapts to using just one of the paths for the whole conversation. The specific selection depends on the network behavior for the specific iteration.

6.4.3 Application Layer Measurements

Now, we tie the results to the robot's performance. Figure 21 shows the accuracy and the trade-off metric of the different configurations in the regular scenario. Focusing on accuracy, we note that the most accurate case is the one with the duplicate threshold 100%, which receives 100% of the actions accurately in most of the iterations. Nevertheless, this case is not shown to be efficient. In fact, it turns out to be the least efficient with a high trade-off metric value. In contrast, the case with a

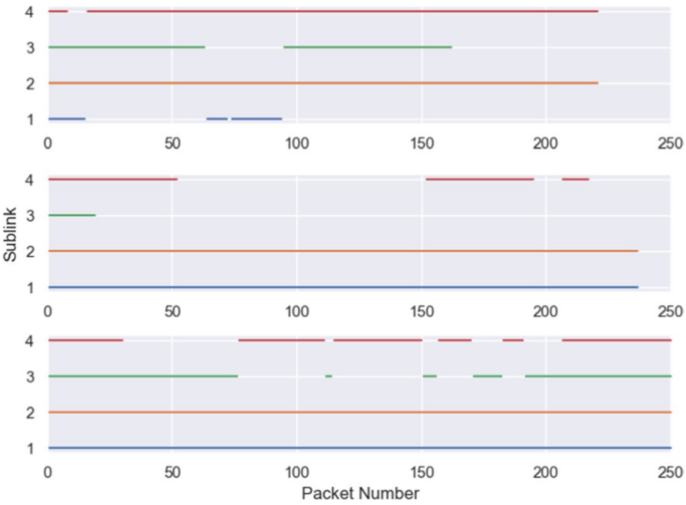


Fig. 19 Example of sublinks used in the impaired scenario

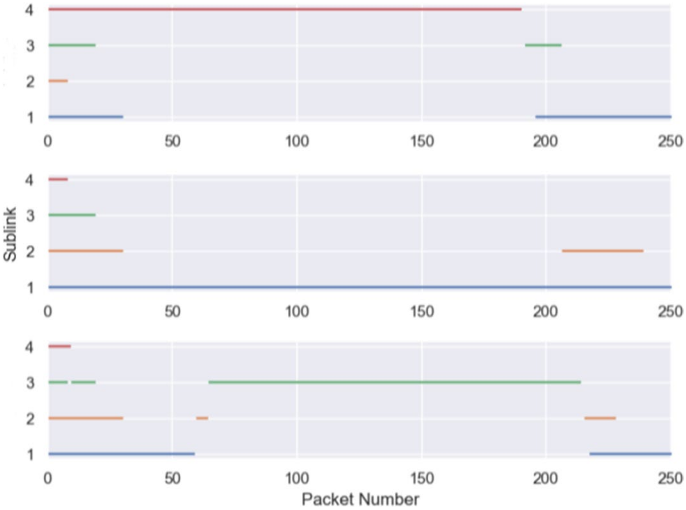
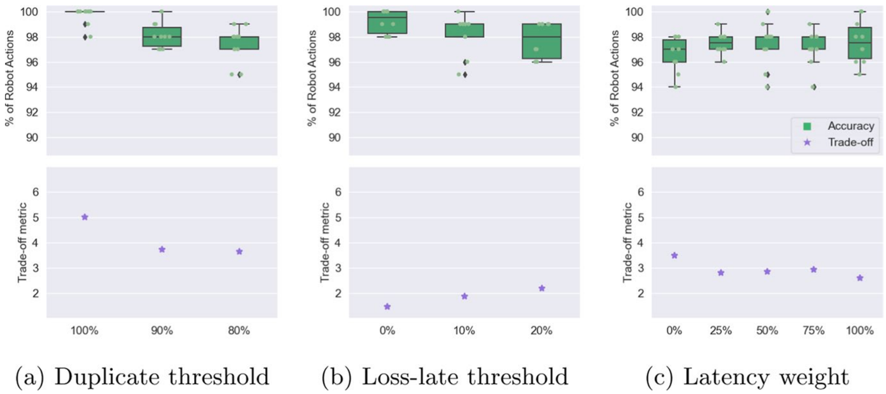


Fig. 20 Example of sublinks used in the regular scenario

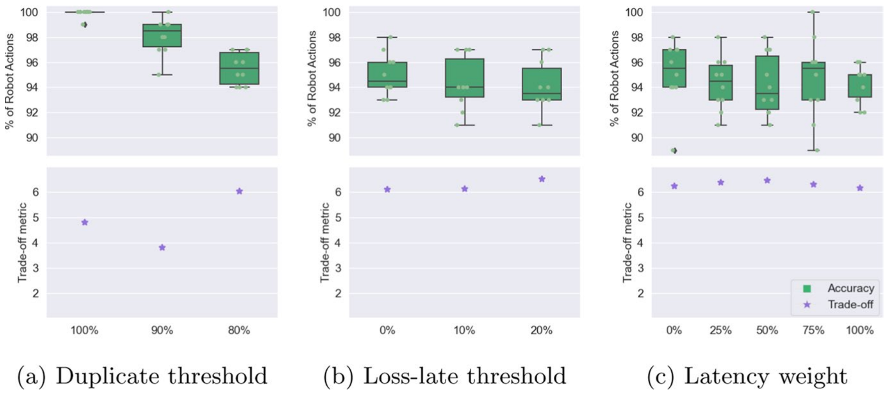
loss-late threshold of 0% shows a good balance between accuracy and efficiency, with more than 99% of mean accuracy and a low trade-off metric, proving to be the most favorable case studied for this scenario.

Figure 22 shows the same evaluation for the impaired scenario. In this case, we can also point to the duplicate threshold 100% as the best scenario in terms of accuracy. This configuration is also one of the best in terms of efficiency. In fact, it is only worse than the duplicate threshold 90% scenario. However, the duplicate threshold 90% scenario shows better efficiency without sacrificing much accuracy



(a) Duplicate threshold (b) Loss-late threshold (c) Latency weight

Fig. 21 Accuracy and trade-off metric in the regular scenario



(a) Duplicate threshold (b) Loss-late threshold (c) Latency weight

Fig. 22 Accuracy and trade-off metric in the impaired scenario

(close to 99% of mean accuracy). Thus, it could be seen as a better configuration for this specific application based on the trade-off metric.

7 Conclusion

In this paper, we evaluate the Multi-connection Tactile Internet Protocol (MTIP), a transport protocol for the remote control of Tactile Internet applications over wireless networks. MTIP uses multiple communication paths or sublinks, with the aim to enhance reliability and meet the latency deadlines expressed by the application through the API. MTIP uses application information and network measurements to select the most suitable sublinks for communication.

We have analyzed the algorithm's behavior to intelligently select sublinks under different configurations of the algorithm parameters. We have also measured the impact of meeting the application demands and the waste of network resources. To validate our results, we have performed a thorough evaluation in diverse scenarios, first in an emulated environment and then in a real 4G/5G lab manipulating an industrial robot.

The results show that more restrictive thresholds reduce the amount of lost and late packets and increase the number of duplicates, while less restrictive thresholds do the opposite. Moreover, regarding the concrete selection of sublinks, we have seen that a proper selection can reduce the number of duplicates, especially in fairly good scenarios. The evaluation in the real environment also shows how MITP can efficiently comply with the demands of a real application in an industrial scenario.

Future work entails exploring new, more expressive API alternatives currently being developed, such as TAPS [42]. Moreover, it will focus on exposing additional parameters to the application, such as protocol configurations, to study the protocol algorithm further and enhance the selection of sublinks.

Author Contributions Conceptualization: [DR, K-JG, AB, PM]; software: [DR]; writing—original draft preparation: [DR]; writing—review and editing: [K-JG, AB, PM]; funding acquisition: [PM]; supervision: [K-JG, AB, PM].

Funding This work supported by the EVOLVED5G project (European Union H2020 research and innovation Grant Agreement No. 101016608), and by the Spanish government (Grant FPU17/04292 and UNICO I+D 5G+TACTILE_1 Project TSI-063000-2021-11). This work was part of the research internship of Delia Rico in Karlstad University, Sweden, supported by the Spanish Ministry of Education through grants EST19/00930 and EST21/00446. Funding for open access charge: Universidad de Málaga / CBUA.

Data Availability Not applicable.

Declarations

Conflict of interest Not applicable.

Ethical Approval Not applicable.

References

1. Holland, O., et al.: The IEEE 1918.1 "Tactile Internet'' Standards Working Group and its Standards. Proc. IEEE **107**(2), 256–279 (2019). https://doi.org/10.1109/JPROC.2018.2885541
2. Postel, J.: User datagram protocol. RFC 768 (August 1980). http://www.rfc-editor.org/rfc/rfc768.txt
3. Postel, J.: Transmission control protocol. RFC 793 (September 1981). http://www.rfc-editor.org/rfc/rfc793.txt
4. Ping, L., Wenjuan, L., Zengqi, S.: Transport layer protocol reconfiguration for network-based robot control system. In: Proceedings of 2005 IEEE Networking, Sensing and Control, pp. 1049–1053 (2005). IEEE
5. Wirz, R., et al.: Efficient transport protocol for networked haptics applications. In: Haptics: Perception, Devices and Scenarios, pp. 3–12. Springer, Berlin (2008)
6. Boukerche, A.F.M., Maamar, H., Hossain, A.: An efficient hybrid multicast transport protocol for collaborative virtual environment with networked haptic. Multimedia Syst. **13**, 283–296 (2006)
7. Cen, A., et al.: Supermedia transport for teleoperations over overlay networks. In: NETWORKING 2005: Networking Technologies, Services, and Protocols. Lecture Notes in Computer Science, vol. 3462, pp. 1409–1412 (2022). https://doi.org/10.1007/11422778_126
8. Dodeller, S.: Transport layer protocols for haptic virtual environments. PhD thesis, University of Ottawa, Canada (2004)
9. Rico, D., Merino, P.: A survey of end-to-end solutions for reliable low-latency communications in 5G networks. IEEE Access **8**, 192808–192834 (2020). https://doi.org/10.1109/ACCESS.2020.3032726
10. Networld2020: The Strategic Research and Innovation Agenda: Smart Networks in the context of NGI. Technical report, European Technology Platform (September 2020). https://www.networld2020.eu/3487-2/
11. Ford, A. et al.: TCP extensions for multipath operation with multiple addresses. RFC 8684 (March 2020). http://www.rfc-editor.org/rfc/rfc8684.txt
12. Iyengar, J., Thomson, M.: Quic: A udp-based multiplexed and secure transport. RFC 9000, RFC Editor (May 2021). https://www.rfc-editor.org/rfc/rfc9000.txt
13. Viernickel, T., et al.: Multipath QUIC: a deployable multipath transport protocol. In: 2018 IEEE International Conference on Communications, pp. 1–7 (2018). https://doi.org/10.1109/ICC.2018.8422951
14. Singh, V., et al.: Multipath RTP (MPRTP). Internet-Draft draft-ietf-avtcore-mprtp-03. IETF Secretariat (June 2016). https://datatracker.ietf.org/doc/draft-ietf-avtcore-mprtp/03/
15. Rico, D., Gallardo, M., Merino, P.: Modeling and verification of the Multi-connection Tactile Internet Protocol. In: Proceedings of the 17th ACM Symposium on QoS and Security for Wireless and Mobile Networks, pp. 105–114 (2021)
16. Rico, D., Grinemmo, K.-J., Brunstrom, A., Merino, P.: Implementation and evaluation of the Multi-connection Tactile Internet Protocol and API. In: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, pp. 1–6 (2022). https://doi.org/10.1109/NOMS54207.2022.9789842
17. Morse Research Group: Morse Lab (2022). https://morse.uma.es/morse-lab/. Accessed 22 Oct 2022
18. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. STD 64 (July 2003). http://www.rfc-editor.org/rfc/rfc3550.txt
19. Stewart, R.: Stream control transmission protocol. RFC 4960, RFC Editor (September 2007). http://www.rfc-editor.org/rfc/rfc4960.txt
20. Vespa, P.M., et al.: Intensive care unit robotic telepresence facilitates rapid physician response to unstable patients and decreased cost in neurointensive care. Surg. Neurol. **67**(4), 331–337 (2007). https://doi.org/10.1016/j.surneu.2006.12.042
21. Dadhich, S., Bodin, U., Andersson, U.: Key challenges in automation of earth-moving machines. Autom. Construct. **68**, 212–222 (2016). https://doi.org/10.1016/j.autcon.2016.05.009
22. Dodeller, S.: Transport layer protocols for telehaptics update messages. In: Proceedings of 22nd Biennial Symposium on Communications, June 2004. https://ruor.uottawa.ca/handle/10393/26627
23. Ping, L., Wenjuan, L., Zengqi, S.: Transport layer protocol reconfiguration for network-based robot control system. In: Proceedings of 2005 IEEE Networking, Sensing and Control, pp. 1049–1053 (2005). https://doi.org/10.1109/ICNSC.2005.1461342
24. Uchimura, Y., Yakoh, T.: Bilateral robot system on the real-time network structure. IEEE Trans. Ind. Electron. **51**(5), 940–946 (2004). https://doi.org/10.1109/TIE.2004.834942

25. Kokkonis, G., Psannis, K., Roumeliotis, M., Kontogiannis, S.: A survey of transport protocols for haptic applications. In: 2012 16th Panhellenic Conference on Informatics, pp. 192–197 (2012). https://doi.org/10.1109/PCi.2012.54
26. Antonakoglou, K., et al.: Toward haptic communications over the 5G Tactile Internet. IEEE Commun. Surveys Tutor. **20**(4), 3034–3059 (2018). https://doi.org/10.1109/COMST.2018.2851452
27. Qadir, J., et al.: Exploiting the power of multiplicity: A holistic survey of network-layer multipath. IEEE Commun. Surveys Tutor. **17**(4), 2176–2213 (2015). https://doi.org/10.1109/COMST.2015.2453941
28. Coninck, Q.D., Bonaventure, O.: Multipath extensions for QUIC (MP-QUIC). Internet-Draft draft-ietf-quic-multipath-04. IETF Secretariat (March 2023). https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/04/
29. Zheng, Z. et al.: Xlink: Qoe-driven multi-path quic transport in large-scale video services. In: Proceedings of the 2021 ACM SIGCOMM 2021 Conference. SIGCOMM '21, pp. 418–432. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3452296.3472893
30. Yang, W., Cao, J., Wu, F.: Adaptive video streaming with scalable video coding using multipath QUIC. In: 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), pp. 1–7 (2021). https://doi.org/10.1109/IPCCC51483.2021.9679445
31. Wu, J., et al.: Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks. IEEE Trans. Mobile Comput. **15**(9), 2345–2361 (2016). https://doi.org/10.1109/TMC.2015.2497238
32. Pakulova, E., Miller, K., Wolisz, A.: Adaptive low-delay video streaming in heterogeneous wireless networks using MPRTP. In: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 14–19 (2017). https://doi.org/10.1109/IWCMC.2017.7986255
33. Shankarkumar V. et al.: Precision Time Protocol Version 2 (PTPv2) Management Information Base. RFC 8173 (June 2017). http://www.rfc-editor.org/rfc/rfc8173.txt
34. Amend, M., Hugo, D.: Multipath sequence maintenance. Internet-Draft draft-amend-iccrg-multipath-reordering-01. IETF Secretariat (November 2020). http://www.ietf.org/internet-drafts/draft-amend-iccrg-multipath-reordering-01.txt
35. The Qt Company: Cross-platform software development for Embedded & Desktop (2021). https://www.qt.io/
36. Mininet Project: Mininet (2021). http://mininet.org/
37. The Linux Foundation: Netem (2021). https://wiki.linuxfoundation.org/networking/netem
38. 3GPP: 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NR; Radio Link Control (RLC) protocol specification (Release 16). Technical Report 38.322, 3rd Generation Partnership Project (December 2020). Version 16.2.0
39. The Linux Foundation: Low latency Linux (2022). https://ubuntu.com/blog/industrial-embedded-systems
40. Clearpath: Husky - Unmanned Ground Vehicle. https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/. Online; accessed 22 Oct 2022 (2022)
41. Open robotics: ROS. https://www.ros.org/. Accessed 22 Oct 2022
42. Pauly, T. et al.: An architecture for transport services. Internet-Draft draft-ietf-taps-arch-16, IETF Secretariat (April 2021). https://datatracker.ietf.org/doc/draft-ietf-taps-arch/16/

Delia Rico obtained a B.Sc. degree in telematics engineering in 2017 and an M.Sc. degree in telematics engineering and telecommunication networks in 2018 from the University of Malaga, Spain. She is currently pursuing her Ph.D. in Computer Science at the same university. She is a member of the ITIS Software Research Institute at the University of Malaga, where she has participated in national research projects such as RFOG or 5G+TACTILE_1 and international projects such as 5Genesis or 6GSandbox. Her research interests include 5G/6G networks, communication protocols, multi-connectivity and deterministic communications.

Karl-Johan Grinemmo received his M.Sc. in Computer Science and Engineering from the Linköping Institute of Technology, Sweden, in 1994. In 2006, he received his Ph.D. in Computer Science from Karlstad University, and since 2010 he has been with the faculty of Karlstad University, Sweden. He became a Senior Lecturer at Karlstad University in the Fall of 2014 and an Associate Professor (Swedish Docent) in the Fall of 2016. He has authored and co-authored more than 80 conference and journal papers and is a Senior Member of IEEE and a member of ACM. His research primarily targets the quality of service in IP-based computer and telecommunication networks. He has worked for almost fifteen years as an engineer in the telecom industry, first at Ericsson and then as a consultant at Tieto. Much of his work was related to Ericsson's signaling system in the mobile core and radio access network. From the Fall of 2009 until the Fall of 2010, he was on leave from Tieto. He worked as an acting Associate Professor at the School of Information and Communication Technology, KTH Royal Institute of Technology.

Anna Brunstrom received a Ph.D. in Computer Science from College of William & Mary, VA, in 1996. She is currently a Full Professor and Research Manager for the Distributed Systems and Communications Research Group at Karlstad University, Sweden. She has a background in distributed systems, but her main area of work over the last years has been in computer networking with a focus on Internet architectures and protocols, techniques for low latency Internet communication, multi-path communication and performance evaluation of mobile broadband systems including 5G/6G. She has authored/coauthored over 200 international peer-reviewed journal and conference papers.

Pedro Merino is a professor at University of Malaga. His research interests are new generation Internet, 5G/6G networks and automated methods for software reliability. He has led more than 40 national and international research projects, most of them in collaboration with industry. He leads a 4G/5G outdoor Victoria Network testbed at Malaga city. He has participated in 15 European projects in the area of mobile networks, with leading role in projects SAID, 5GENESIS, EuWIRELESS and 6G-SANDBOX. He is the director of the Institute for Software Engineering and Software Technology (ITIS Software) at University of Malaga.