Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Medientechnik

A Multiplexing Scheme for Multimodal Teleoperation

Burak Çizmeci

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		apl. Prof. Dr.-Ing. Walter Stechele
Prüfer der Dissertation:	1.	Prof. Dr.-Ing. Eckehard Steinbach
	2.	Prof. Çağatay Başdoğan, Ph.D.

Die Dissertation wurde am 04.04.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 24.07.2017 angenommen.

To all those who made this happen.

Abstract

Teleoperation systems give us the ability to immerse ourselves in environments that are remote or inaccessible to human beings. Teleoperation systems are also referred to as telemanipulation systems, considering their ability to provide manipulations in a remote environment. In addition to auditory and visual feedback, the bidirectional exchange of haptic information enables the human operator to interact physically with the remote objects. Throughout the years, in haptics and control engineering fields, researchers have focused on the development of stable and transparent teleoperation systems providing remote touch functionality. The major goal is to present high-quality kinesthetic feedback in a bilateral control loop closed by a human operator and a network that introduces delays and losses to the exchanged signals. On the other hand, communicating multimodal signals, such as video, audio and haptics, demands a high bitrate to facilitate remote manipulation. Very few frameworks investigating the communication issues of a multimodal teleoperation system have been developed.

This thesis introduces a complete setup of a teleoperation system using commercially available robotic devices and computer hardware and proposes a novel communication protocol as an application layer multiplexing scheme for multimodal signals, which is embedded into the bilateral control loop of a teleoperation system. The multiplexing scheme applies preemptive-resume scheduling to stream audio and video data by giving high priority to haptic signals. Meanwhile, a real-time transmission rate estimator is implemented for the prediction of the transmission capacity in order to adapt the video bitrate and data throughput rate. Moreover, a detection scheme for unexpected transmission rate drops is introduced to make the system resilient to such abrupt changes in the network. Additionally, low-delay video communication is critical for a teleoperation system to achieve good-quality and fast visual feedback. Consequently, a video encoder with an accurate bitrate controller is developed to stream the compressed teleoperation scenes fitting into the target bitrate. The performance of the overall system is thoroughly evaluated using objective metrics to monitor the end-to-end delay provided by the system. According to the evaluations, the multiplexing scheme can guarantee end-to-end delays under a variety of network conditions, and the teleoperation system is resilient to unexpected changes in the available transmission rate.

Kurzfassung

Teleoperationssysteme ermöglichen es einem Benutzer Umgebungen, die fern oder unzugänglich für Menschen sind, zu erreichen. Teleoperationssysteme werden auch als Telemanipulationssysteme wegen ihrer Fernwartungsfähigkeiten bezeichnet. Zusätzlich zu akustischem und visuellem Feedback, kann der menschliche Operator durch dem Austausch haptischer Information ferne Objekte berühren und manipulieren. Während der letzen Jahre hat sich die Forschung im Bereich Haptik und Regelungstechnik auf die Entwicklung von stabilen und transparenten Teleoperationssystemen mit Berührungsfeedback konzentriert. Das wichtigste Ziel ist dabei, realistische kinästhetische Rückkopplung in einem bilateralen Regelkreis, einschließlich menschlichem Operator in einer Netzwerkumgebung mit Zeitverzögerung und mit Datenverlust, zu ermöglichen. Hinzu kommt noch die Realisierung der Übertragung von multimodalen Signalen wie Video, Audio und Haptik, die hohe Datenraten erfordern um die Interaktion in der Ferne zu ermöglichen. Es existieren bisher nur wenige Lösungsansätze welche die Kommunikationsherausforderungen eines multimodalen Teleoperationssystems addressieren.

Die vorliegende Doktorarbeit stellt ein vollständiges Teleoperationssystem mit Roboter und Computerhardware vor, die es auf dem Markt bereits zu erwerben gibt. Basierend darauf wird ein neuartiges Kommunikationsprotokoll vorgeschlagen, das einen speziell auf die Anforderungen der Teleoperation abgestimmten Multiplexer enthält. Der Multiplexer ist in dem bilateralen Regelkreis des Teleoperationssystems integriert und verwendet *Preemptive-Resume Scheduling* zum Streamen der Audio- und Videodaten einschließlich hochpriorisierter haptischer Signale. Daher wurde eine Echtzeit-Bandbreitenmessung implementiert, um die Vorhersage der verfügbaren Übertragungskapazität zu ermöglichen. Danach wird nach Bedarf die Bitrate des Multiplexers angepasst. Darüber hinaus wurde eine Methode entwickelt, die einen abrupten Bandbreitenabfall erkennt, da sonst das System gegenüber solchen Netzwerk-Schwankungen anfällig wäre. Hinzu kommt, dass eine Videokommunikation mit niedrigem Zeitverlust für das Erreichen einer visuellen Rückkopplung in guter Qualität notwendig ist. Daher wurde ein Videoencoder mit einem genauen Bitraten-Controller entwickelt, um eine Ziel-bitrate einzuhalten. Die Leistungsfähigkeit des Gesamtsystems wird mit Hilfe von Ende-zu-Ende-Verzögerungsmessungen objektiv untersucht. Nach den Latenz-Bewertungen kann

der Multiplexer den Einsatz in verschiedenen Netzwerkumgebgungen garantieren, ohne dass eine Teleoperation bei unerwarteten Einbrüchen der Übertragunsrate gestört oder gar ganz unterbrochen wird.

Acknowledgements

This dissertation was produced as a member of the research and teaching staff at the Chair of Media Technology (LMT) at the Technical University of Munich. My research activities at LMT were supported initially by the German Academic Exchange Service (DAAD) and later by the European Research Council under the European Union's 7^{th} Framework Programme (FP7/2007-2013)/ERC Grant agreement no. 258941. To make this great work happen, many people have supported me morally, personally and professionally. I am genuinely grateful to all of them. I ask forgiveness to people who are not explicitly mentioned here. First, I would like to show my special gratitude to my Ph.D. supervisor, Prof. Dr.-Ing. Eckehard Steinbach for providing me with the opportunity to conduct research in such a productive and innovative group. He encouraged me to move to haptics research, which I realize now was a very important decision in my academic career. Compared to many Ph.D. students around the world, I feel very lucky that I had the chance to work with the most up-to-date and best equipment in my research field. This occurred through the gradual contribution of my supervisor and his former Ph.D. students to the Haptic Communication research over the years. I would like to thank Dr.-Ing. Peter Hinterseer, Dr.-Ing. Julius Kammerl, Dr.-Ing. Rahul Chaudhari and Fernanda Brandi as the initial contributors to kick off the project.

Building this framework was not straightforward. Almost every one at LMT provided suggestions and ideas to implement a real teleoperation system in our chair. For the construction of the robotics lab, substantial man power and motivation were needed to set up such a stable and secure system. I would like to thank my colleagues Dr.-Ing. Clemens Schuwerk, Dr.-Ing. Nicolas Alt, Dr.-Ing. Julius Kammerl and Dr.-Ing. Rahul Chaudhari, who really put significant effort into building the lab. I would like to sincerely thank Dr.-Ing. Rahul Chaudhari for his support and significant contributions to the development of the multiplexing scheme. He helped me a lot in making the transition to research in haptics. The video compression part of the project was implemented in collaboration with two people. I would like to thank my student Michael Eiler, who was assigned to the project as his Bachelor's Thesis; his objective was to integrate the original ρ-domain RC scheme into the end-to-end real-time streaming test environment. He implemented the base of the software framework using an ethernet-based GigE camera and two computers physically separated by a network emulator. Furthermore,

he accelerated the mathematical computations of the scheme. I would also like to thank Min Gao, who was a visiting researcher for 6 months during his Ph.D. studies; he was assigned to work with me to improve the RC scheme. We worked together closely and developed the exponential (ρ,QP) model to accelerate the RC scheme. In addition, he added new features to the scheme such as the MB-level rate allocation and the smooth QP determination for neighboring MBs. Additionally, I would like to thank Dr.-Ing. Fan Zhang, who was the first implementer of the RC scheme in our lab. As an initiator of the project, he enlightened us on possible improvements for achieving a low-delay visual communication system.

I would also like to thank Robert Huitl for his kind help on programming issues that we encountered during the development of the system. I would like to thank Dr.-Ing. Nicolas Alt for his contributions to the development of the drivers to control the KUKA lightweight robot arm. Due to his support, the implementation time of the framework was greatly reduced. I would like to especially thank Xiao Xu for contributing significantly to the development of the time-domain passivity control architecture with haptic data reduction schemes. Due to his support, we could investigate more realistic network situations for a teleoperation system. I would like to thank Christoph Bachhuber for the development of the delay measurement system for visual communications. Due to his support, I could precisely measure the visual delay of the teleoperation system. I would like to thank Dr. Giulia Paggetti for collaborating on the psychophysical studies that we performed together for teleoperation systems. It was a very nice experience to work with a psychologist. I would like to also thank Alexandra Zayets for taking over the teaching activities in the image and video compression course. Due to her help, I could better concentrate on my thesis. Many thanks go to all of my colleagues at LMT, especially Anas Al-Nuaimi, Dr.-Ing. Jianshu Chao, Dr.-Ing. Ali El Essaili, Dr.-Ing. Hu Chen, Dr.-Ing. Werner Maier, Dr.-Ing. Florian Schweiger, Fernanda Brandi, Damien Schröder, Dominik van Opdenbosch, Matti Strese, Jingyi Xu and Tamay Aykut for their heartfelt friendship and enjoyable times that we spent together. I would like to give special thanks to Ingrid Jamrath, Dr. Martin Maier, our beloved secretary Gabriele Kohl, who passed away in 2015, Marta Giunta and Brigitte Vrochte for their reliable administrative support. I would like to give special thanks to Simon Krapf as well for his very kind technical support. Overall, I can say that I have gained substantial experience in this multinational research group. I would like to show my appreciation to all of my students who worked with me over the years. In particular, I would like to thank Cem Dillioğlugil, who helped me implement the psychophysical tests for Dr. Giulia Paggetti. I would like to give special thanks to Yiğit Özer and Mustafa Tok for helping me develop an end-to-end low-delay video streaming system.

Finally, I am very grateful to all members of my family, especially my brother Kerem Çizmeci who helped me to design the experimental platform, my grandparents, parents and sister for motivating and supporting me during my Ph.D. studies.

Contents

Chapter 1

Introduction

Since the 19^{th} century, transferring human sensory information to a remote location has played a key role in the development of information technologies. The initial point was the invention of the telephone by Alexander Bell and Thomas Watson in 1876, which allowed us to exchange voice signals of a conversation from a distant location. Simultaneously, enormous developments were being achieved in the wireless signal transmission field using electromagnetic waves, which led to the invention of radio and television broadcasting technologies at the end of the 19^{th} century. The dominating senses of communication for information technologies have been auditory and visual stimuli. Because audio-video communication techniques have reached a mature state, they have become an indispensable part of our daily lives. In particular, videoconferencing and live media streaming have become the major communication tools for social and business interactions today.

On the other hand, in the mid 20^{th} century, the space exploration missions started the exciting journey of humans being in space. As a result of the space programs, research and development activities have been significantly accelerated in many scientific disciplines. In particular, wireless communications and robotics technologies have reached a mature state. During space missions, remote manipulation is an essential approach in preserving human life in dangerous situations and for performing tasks in inaccessible locations. Therefore, remote manipulation (telemanipulation/teleoperation) using robots has become a target application for reducing hazards to humans and saving time and money as well. Performing telemanipulation brings about new engineering challenges such as the communication of sensory data and the design of a stable control system for the robotics hardware. In this context, the focus of this thesis is on the study of the communication of audio, video and haptic signals for teleoperation systems.

The term *haptics* is considered to refer to any type of interaction involving the human sense of touch. Recent achievements in robotics and sensor-actuator technologies have brought

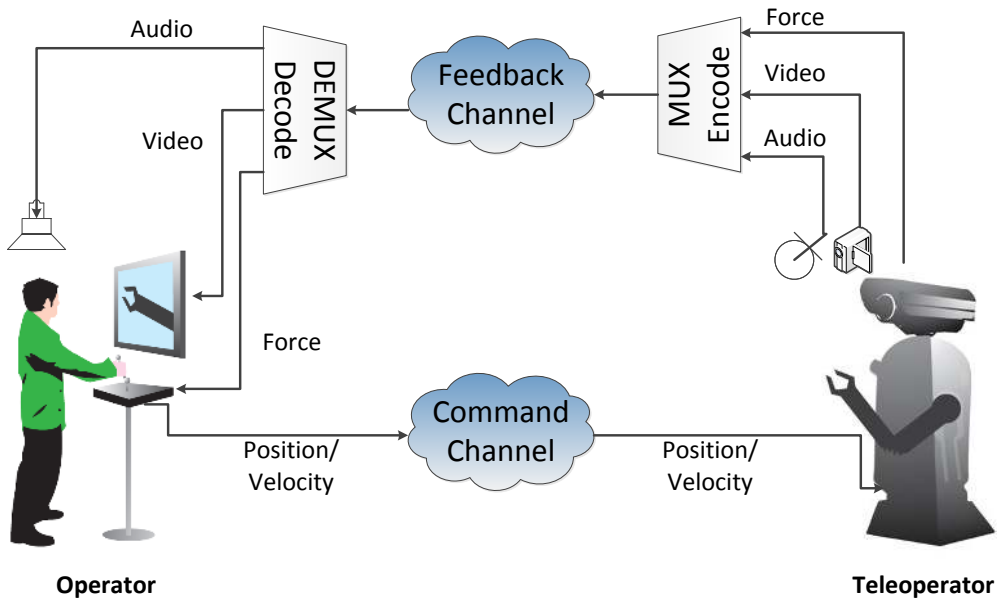


Figure 1.1: A bilateral multimodal teleoperation system. The human operator is connected to the remotely located teleoperator via command and feedback channels. The interaction signals are multiplexed and transmitted over the feedback channel. The received bitstream is demultiplexed, and each modality is displayed to the human operator. In the reverse direction, the position/velocity commands of the human operator are transmitted over the command channel, and the teleoperator joints are moved to reach the target location in the remote environment.

about the possibility to communicate haptic information in addition to audio-visual content. The feeling of touch allows us to interact *physically* with remote objects and perform distant manipulation tasks. A *haptic teleoperation* system can be considered as the main challenging application type involving haptic communications. As observed in Fig. 1.1, a haptic teleoperation system consists of a *human operator* with a *master haptic device*, a *slave robot manipulator* and a *transmission medium* providing interactive communication between the *operator* and the *teleoperator* [Fer65]. The human operator (OP) directly controls the haptic device, which captures position/velocity commands and displays force/torque feedback coming from the remote interaction. At the teleoperator (TOP) side, the audio-visual signals and force/torque feedback are captured, multiplexed and sent to the OP. In parallel, the received velocity/position command is applied at the end-effector of the manipulator. In this context, the communication medium takes on the important responsibility of exchanging the signal flow between the OP and the TOP in real time by providing the lowest possible latency conditions. Haptic teleoperation systems can be applied to many practical scenarios that involve distant manipulation tasks in environments that are inaccessible or dangerous to human beings. Such applications exist in medical areas, including telesurgery and minimally

invasive surgery [Tav08]; in space exploration [She93], including the on-orbit servicing of satellites [GOU+12]; and military tasks such as mine cleaning [KCL+03]. Furthermore, artificial manipulation systems involving haptic feedback [HBS99] have been developed to imitate real environments such as for medical training [TC97], flight simulation [Lof94] applications, video games and entertainment [OTT+95] and education purposes [Sul15].

Since 1965, with the first publication being from Ferrel, researchers in control engineering have been studying haptic telemanipulation and teleaction systems to achieve reliable and stable remote operation while communicating the signals over a network introducing bidirectional delay between the OP and TOP. In his paper [Fer65], he introduced the first force feedback system over a network with bidirectional delay and demonstrated the instability issues of controlling a slave remotely. To provide stable teleoperation, passivity-based control architectures that rely on scattering transformations (also known as wave-variable transformations) were first proposed in [AS89b] by Anderson and Spong. In [NS91], Niemeyer and Slotine extended the wave-variable approach for passive teleoperation over a network introducing bidirectional delay between the OP and TOP. In their solution, they theoretically showed that it is possible to achieve a stable teleoperation system if the delay between the OP and TOP is known and constant over time. In [HR01] and [RKH04], Hannaford and Ryu first introduced the time-domain passivity method to stabilize haptic interfaces. In their approach, they defined the passivity observation concept to balance the incoming and outgoing energies in the system. The passivity controller dissipates/damps the extra energy to ensure the stability of the system. In [RAP10], the authors extended the time-domain passivity approach for haptic teleoperation systems that run over communication links with latency.

Separating the OP and the TOP with an unreliable network, such as the internet, encumbers the control architecture with the task of addressing time-varying delay and packet loss conditions. Extended versions of scattering-transformation-based control architectures [CBS08] were proposed to address the time-varying delay and packet loss issues. In [RAP10], Ryu et al. also showed that the time-domain passivity controller performs well in the presence of time-varying delay and black-out conditions. Although the passivity-based control solutions provide passive and stable teleoperation, they reduce the system transparency with adaptive damping, which reduces the intensity of the force feedback signal. The model-mediated teleoperation (MMT) approach represents an alternative solution that can guarantee both system stability and transparency in the presence of arbitrary communication delays [MN08, FPB10] and packet loss [WBBN12]. The MMT approach employs a local object model at the OP side to approximate the slave environment. The model parameters represent the object in the TOP environment, and they need to be continuously estimated and transmitted back to the OP whenever the TOP predicts a new environment model. At the OP side, the local environment model is reconstructed based on the received model parameters. Then, the haptic

feedback can be rendered depending on the local model without noticeable delay. The model extraction is the crucial part of MMT and is challenging under dynamic environments. For static environments, both stable and transparent teleoperation systems can be achieved if the estimated model is an accurate approximation of the remote environment [PPB10b, PPB10a].

Another stability constraint on the control loop of a haptic teleoperation system is the high sampling rate of signals. The haptic signal acquisition and exchange rates need to be 1 kHz or even higher [CB94]. Hence, a typical haptic teleoperation system needs to transmit and receive data at a rate of 1000 $packets/second$ over the communication medium. However, today's communication networks, such as wireless links and the internet, remain error prone and can cause high latencies and data losses due to the heavy loads of transmission traffic and limited capacities [SHE+12, BPB+13]. In the literature, various authors have proposed new methods to reduce the 1000 $packets/second$ requirement using lossy haptic data reduction schemes. Hirche and Hinterseer et al. proposed the first sample-based haptic data reduction schemes in [HHSB05, HSHB05]. In [KKHB06], the authors further investigated the performance of several frame-based and sample-based haptic data reduction schemes by ensuring the passivity conditions and showed that sample-based data reduction methods achieve better immersion performance than do frame-based approaches. In [HHSB05, HHSB07], Hirche et al. proposed a deadband-based haptic data reduction scheme that irregularly samples the haptic signals by applying a certain deadband threshold. If the current sample exceeds the deadband width, the sample is transmitted. In [HSHB05, HHC+08], Hinterseer et al. extended the deadband approach in [HHSB05, HHSB07] with perceptual thresholds by applying the deadband thresholds using the Weber fraction [Web51]. In these sample-based methods, the haptic signals are locally upsampled back to the original rate, such as 1 kHz, to keep the local control loops stable. However, the data reduction schemes need to be modified to ensure the stability when there is communication delay. In [HB07], Hirche and Buss combined deadband data reduction [HHSB07] with a wave-variable-based control architecture [NS91]. To achieve a better perceptual performance, Vittorias et al. changed the cascaded order of data reduction and scattering transformation in [HB07] by locally computing the wave-variables. Using this approach, perceptual deadband-based data reduction [HHC+08] can be applied in the time domain on haptic signals. The drawback of this method is that the network delay should be constant and known by the system. In [XCSS15], Xu et al. successfully integrated the time-domain passivity control in [RAP10] and the perceptual deadband-based haptic data reduction in [HHC+08] to achieve a stable and perceptually high-quality force-reflecting teleoperation under time-varying delay conditions.

Researchers working in haptics with a focus on the network engineering domain mainly concentrate on improving or developing transport-layer protocols that give high priority to the applications containing haptic interactions. In [PWZ05], the authors suggested a new real-

time protocol that determines the required bitrate based on the content. If the stream includes an interactive application, it receives more bitrate resources compared to non-interactive applications. In [CML+05], the authors focused on building a teleoperation system over overlay networks. In their approach, QoS management is performed between multimodal signals of a teleoperation session at the transport layer by applying rate shapers. In [CMZX05], the authors studied intelligent relaying based on the changing QoS conditions of several paths in the network.

Researchers in the signal processing domain have worked on streaming the multimodal signals of haptic-involved systems and have developed application layer protocols to control the QoS constraints and bitrate resource allocation between modalities. In [OEIS07], the authors proposed an application layer protocol for haptic networking (ALPHAN) that is embedded on top of UDP. Instead of using RTP, ALPHAN introduces its own specific headers related to the haptic interaction, which reduces overhead because ALPHAN transmits its packets at a 1 kHz rate. Additionally, a specific object of an application can be prioritized based on a buffering scheme. In [CSKR07, CHK+09], instead of a new protocol design, the authors multiplexed haptic content into MPEG-4 BIFS (binary format for scenes) and developed a multimodal broadcasting scheme for applications involving passive sense of touch. In [ITN11, KNT15], the authors studied application layer media buffering and skipping techniques for the low bitrate communication of multimodal streams. In [ECES11], the authors extended the ALPHAN protocol with a statistical multiplexing scheme that allocates the available transmission rate resources to audio, video and haptic data based on defined QoS constraints. In [YTY13, YYYK14], the authors applied an end-to-end flow controller to adapt the throughput of the system with source skipping.

Time-varying delay, packet loss and high data rate issues represent the current main challenges of haptic teleoperation systems. Researchers in control engineering have worked hard to develop complete end-to-end force-reflecting systems that run over time-delayed networks. On the other hand, better quality-of-service conditions for haptic teleoperation can be provided from signal processing and network engineering perspectives. In particular, the visual communication part of teleoperation systems demands greater network transmission rate, and the video stream may block the haptic signals over the network due to the network's limited capacities. This is one of the main reasons for time-varying delay issues in haptic teleoperation systems, which has not been comprehensively studied. For interactive streaming applications, such as teleoperation, online gaming and videoconferencing, constant bitrate (CBR) control for video communication needs to be employed to prevent unexpected visual delays because of the transmission rate constraint. The rate control should be handled at the frame level or even at the macroblock (MB) level to provide low delay for each frame. There are many rate control approaches in the literature. In this thesis, we employed a well-known and widely used

video compression standard, H.264/AVC (advanced video codec) [ITU05], and improved the rate control scheme for low-delay teleoperation applications. Therefore, we also focus on rate control techniques that can be applied for the H.264/AVC standard. In [MGWL03], Li et al. proposed a rate control algorithm based on a quadratic model for the rate-quantization (R-Q) relationship, which was later adopted in the reference implementation of the H.264/AVC codec [Joi]. In [LGP+06], the authors extended the quadratic model for an MB-level rate control scheme, which was also adopted in the H.264/AVC reference software. To improve the performance of [LGP+06] Jiang et al. developed more accurate frame-level bit allocation and mean absolute difference (MAD) estimations in [JL06]. To improve the model parameter estimation accuracy, a linear R-Q model-based MB-level rate control was proposed in [DL09] using a context adaptive prediction scheme. However, these algorithms occasionally suffer from large errors in the bitrate estimation due to inaccurate source models. In [HM02b], the authors showed that the bitrate R follows a linear relationship with ρ, which is defined as the percentage of zero transform coefficients after quantization. This linear model between R and ρ has been exploited for rate control in H.263 and MPEG-4 [HKM01, HM02a], and the model can achieve more accurate bitrate estimation. Later, a ρ-domain rate control scheme was proposed in [HW08] with a two-loop encoding pipeline, in which the frame-level statistics are collected in the first loop and used in the second loop to determine the proper quantization parameter (QP) for each MB. An improved ρ-domain rate control was proposed in [ZS11] with a more accurate header bit estimation. However, it is not easy to find a one-to-one mapping between ρ and QP due to the complicated coefficient quantization scheme in H.264/AVC [KSK07]. In [HW08] and [ZS11], the transform coefficients are quantized using all possible QPs to obtain the mapping between ρ and QP. Then, the mapped (ρ,QP) pairs are searched to find the proper QP for the target bitrate R. To reduce the complexity, a linear model was proposed in [LGLC10] to determine the relationships among QP, the frame complexity represented by MAD and ρ. However, this model is not sufficiently accurate at the MB level and may induce large errors in the bitrate estimation. As part of this thesis, an accurate exponential model for ρ and QP has been developed in [GCE+15], which is later discussed in Chapter 4.

This thesis proposes an application layer multiplexing scheme that provides guaranteed transmission of multimodal streams over bandlimited networks with time delay. In contrast to the aforementioned related studies [OEIS07, ECES11, ITN11, YTY13, YYYK14, KNT15], the framework in this thesis employs advanced audio-video compression schemes with rate-shaping capabilities [ITU05, MV06, VTMM10, GCE+15] and state-of-the-art haptic data reduction methods [HHC+08, XCSS15]. Furthermore, we introduce a real teleoperation test environment using commercially available robotics hardware. In contrast to the many control engineering studies related to teleoperation, using consumer hardware when building the

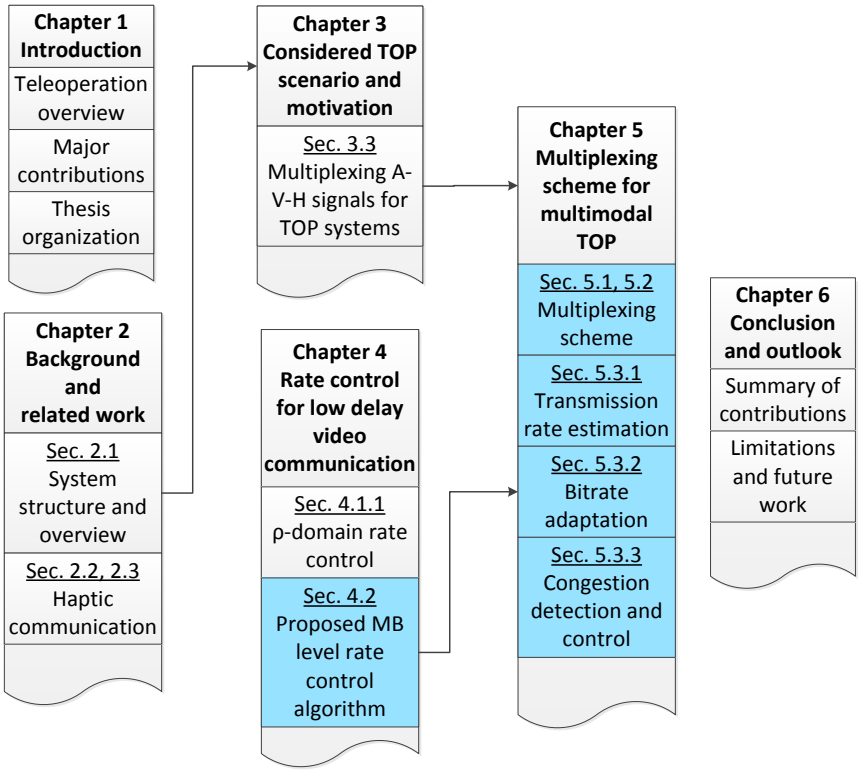


Figure 1.2: Overview of this thesis. The turquoise colored boxes emphasize the original contributions of this thesis. In Section 3.3, the motivation of the multimodal multiplexing scheme for teleoperation systems is introduced, and the detailed design of the scheme is discussed in Chapter 5. In Chapter 4, the MB-level rate control algorithm for low-delay video communication is introduced, and further extensions for bitrate adaptation are discussed in Section 5.3.2.

teleoperation system makes this framework appealing to both academia and industry because the teleoperation setup can be replicated anywhere for research and commercial needs. The proposed multiplexing scheme can be considered as a software package that handles the bidirectional communication of multimodal streams using PCs for the teleoperation setup.

In Fig. 1.2, the organization of the thesis is given. In Chapter 1, a detailed literature review of teleoperation systems is provided. Chapter 2 introduces the teleoperation system structure and its building blocks. In Chapter 3, the considered teleoperation scenario is introduced, and the necessity of applying an application-layer multiplexer for audio, video and haptic signals is discussed. In Chapter 4, the rate control scheme developed for the video communication part of the system is introduced. In Chapter 5, the proposed multiplexing scheme for multimodal teleoperation is introduced, and comprehensive experimental results are discussed in detail. Finally, Chapter 6 concludes the thesis with some discussions on the contributions of this thesis, limitations and future work.

Major contributions and thesis organization

This thesis addresses the effects of delay caused by the available low transmission rate of the communication link between the TOP and OP and proposes video encoding and network engineering solutions to provide a guaranteed end-to-end delay for teleoperation systems running with audio-video and haptic modalities. The major contributions can be summarized as follows:

- **Low-delay video communication:** The visual feedback streaming of the teleoperation system needs to be as fast as possible to ensure the real-time manipulation of remote objects. The capacity of the communication link plays a critical role in the delay of video frames. Hence, the bitstream of the video frames should fit into the communication bottleneck to satisfy both the visual quality and delay requirements. In Chapter 4 Section 4.2, we present a macroblock (MB)-level rate control algorithm for low-delay H.264/AVC video communication based on the ρ-domain rate model [HM02b], which relies on the linear relation between the percentage (ρ) of zero discrete cosine transform (DCT) coefficients after quantization within a frame and the overall bitrate. In the proposed algorithm, an exponential model is used to characterize the relation between ρ and the quantization step ($Qstep$) at the MB level, with which the quantization parameter (QP) for an MB can be obtained. Furthermore, a switched QP calculation scheme is introduced to obtain the QP for each MB to avoid large deviations of the actual frame size from the target bit budget. The proposed MB-level rate control algorithm is compared with the original ρ-domain rate control by applying objective evaluation metrics: the peak-signal-to-noise ratio (PSNR) for visual quality, the bitrate accuracy for rate control performance, and the percentage of encoding time reduction for computational complexity. Further online delay-jitter tests are performed over a low-capacity channel to illustrate the real-time performance of the proposed low-delay video communication system in comparison with the original rate control of the x264 codec [MV06, MV07].

- **Multiplexing scheme for multimodal teleoperation:** If the transmission rate is low and limited, the multimodal signals compete for the narrow capacity. In this case, the signals need to be scheduled at the application layer based on their priorities. In Chapter 5 Section 5.1, a novel multiplexing scheme [CCX+14] is proposed for multimodal teleoperation over communication links with known or estimated capacities. The multiplexing approach uniformly partitions the channel into 1 ms resource buckets and applies a buffered preemptive and resume scheduling approach by controlling the size of the transmitted video packets. The preemptive and resume decisions are made based on the irregular haptic transmission events generated by the perceptual haptic data reduction approach [HHC+08, XCSS15]. The multiplexing scheme proposed in [CCX+14] is extended in [CXC+17] with the following features:

- **Channel adaptive streaming for teleoperation:** The correct estimation of the network transmission rate plays a critical role in the performance of a teleoperation system. Especially for low-bitrate cases, the system needs to utilize the available capacity efficiently considering the quality of the signals and target delay constraints. In Chapter 5 Section 5.3, we discuss the difficulties of the transmission rate estimation problem and introduce the development and adoption of a TCP-based flow control algorithm [CFM04] into the teleoperation system. Since the system communicates over UDP due to real-time requirements, an acknowledgment mechanism is added to the demultiplexer side of the scheme to estimate the transmission capacity of the link. Although the transmission rate estimator [CFM04] performs efficiently based on simulation results, applying it to a real and interactive system brings additional implementation-related challenges. For instance, the packet processing loops at the application layer must run as fast as possible to sample the time to ensure the precision of the transmission rate estimation. Therefore, the math computations and memory operations are accelerated using assembly versions of the functions and fixed-point arithmetic. In this thesis, we assumed that the transmission link between the OP and TOP does not apply any priority to teleoperation streams, and it does not include a feedback scheme to warn the teleoperation system about congestion and transmission rate changes. To evaluate the performance of the teleoperation system, the following transmission rate conditions are tested:

  CBR links: The available transmission rate between the OP and TOP is constant over time, and it is estimated by the system to automatically adjust the system throughput rate, video bitrate and multiplexing buffer based on the link capacity. In this experiment, 1, 2 and 3 $Mbps$ CBR links are tested.

  Time-varying transmission capacity: The available transmission rate varies over time, with a mean bitrate of 1.2 $Mbps$ and a standard deviation of 95 $kbps$. This experiment challenges the system with sudden transmission rate rises and drops to investigate the system response for the transmission rate estimation and delay-jitter of the signals.

  CBR link shared with another session: A 4 $Mbps$ CBR link is shared with another TOP session producing a mean bitrate of 2 $Mbps$. This experiment studies the system response if the transmission medium does not apply any scheduling discipline on the incoming TOP streams.

  CBR link with congestion: The available constant transmission rate drops suddenly from 3 $Mbps$ to 2 $Mbps$ due to a rescheduling in the transmission medium because the service provider allocates bitrate resources to another bitrate-demanding application. This experiment illustrates the challenge of detecting congestion events and estimating the transmission rate when the delay between the OP and TOP increases.

Furthermore, the following control mechanisms are added to regulate the incoming video traffic at the multiplexer input, which must fit within the current transmission rate of the communication link. To handle the adaptation, the multiplexer communicates with the video encoder and updates encoding parameters for the current conditions.

(1) The low-delay video communication system (Chapter 4) is equipped with a bitrate-controlling algorithm that ensures accurate frame-level rate control. Whenever a new transmission rate estimation is made, the multiplexer updates the video bitrate immediately. However, there is also incoming traffic to the multiplexer from audio and haptic signals. Therefore, this overhead needs to be predicted to ensure correct adaptation. Section 5.3.2 proposes a linear model to handle the internal side traffic. Furthermore, a single-frame delay constraint [DvBA08] is applied to the final bitrate of the video stream to guarantee a constant delay for the visual feedback.

(2) The transmission rate estimation algorithm [CFM04] is very sensitive to network capacity changes. On the other hand, the round-trip time (RTT) delay (or the two-way propagation delay) impairs the estimation and leads to the lagged estimation of transmission rate during congestion events. In [CFM04], the transmission rate estimation algorithm is tested with a maximum RTT of 100 ms. However, RTT delays of greater than 100 ms can be encountered for geographically distant and space exploration teleoperation sessions, where the system clearly fails. Therefore, the system should be resilient to sudden capacity drops in the network. Section 5.3.3 introduces a congestion control scheme to converge to the true network capacity quickly once congestion is detected. The estimated transmission rate is tracked over time, and the scheme detects sudden congestion events to adapt the system parameters to the current network conditions. During the congestion event, the scheme switches to congestion control mode to converge smoothly to the current network transmission rate. For the congestion control mode, the multiplexer uses its communication to the video encoder for adapting the frame types (I and P frames), frame rate and bitrate.

Parts of this thesis were published in various international peer-reviewed scientific journals and conferences [BCS12, CCK+12, XCS13, CCX+14, PCDS14, XCANS14, GCE+15, XCSS15, XSCS16, XCSS16, CXC+17].

Chapter 2

Background and Related Work

Haptic teleoperation systems

Haptic teleoperation systems immerse the human operator (OP) into remote environments that, for human beings, are distant, inaccessible, scaled or dangerous for performing manipulation tasks. In the following subsections, the building blocks of the teleoperation system developed in this thesis are introduced in detail.

2.1 System structure and overview

A typical teleoperation system consists of three main components, as shown in Fig. 2.1. The human OP interacts with the remote environment through a human-system interface (HSI), a teleoperator (TOP) robot and a communication network, which exchanges the signals between them.

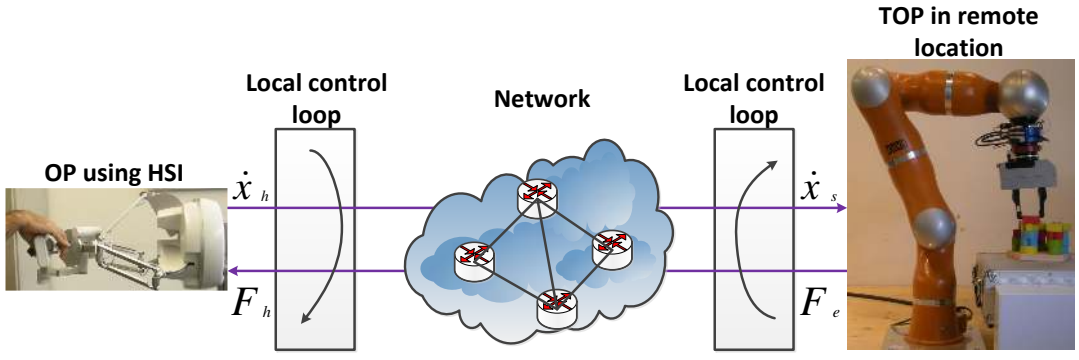


Figure 2.1: General structure of a typical haptic teleoperation system: the HSI, the communication medium and the remotely located teleoperator (reproduced from [CAS14]).

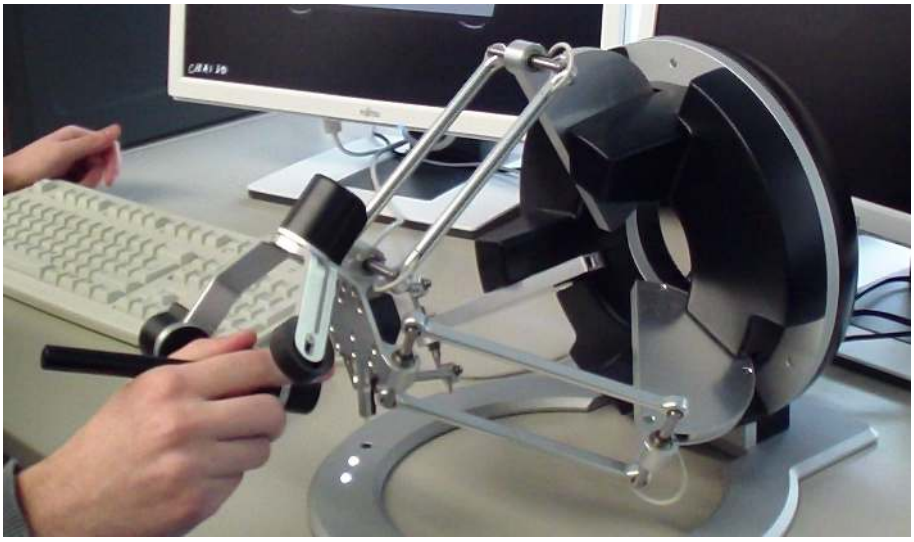


Figure 2.2: Force Dimension Omega 6 Haptic Device: A typical force-reflecting device, as shown here, captures human position and rotation commands via servo encoders and, as a feedback, provides a single-point contact force to the human hand through the servo motors.

2.1.1 Human-system interface

The HSI is composed of a haptic device for receiving haptic feedback and sending the position/velocity commands to the robot, a video screen for displaying visual feedback and headphones for listening for the audio feedback from the remote environment. The HSI can be extended with additional modalities as the sensor and actuator of the modality are integrated into the TOP and HSI, respectively. In this setup, we focused on displaying three major modalities: kinesthetic force and visual and auditory feedback. A Force Dimension Omega 6 [For01] (see Fig. 2.2) is used as the haptic interface to exchange the force feedback and position/velocity signals of the human OP. The Omega 6 is a high-precision haptic device that has 6 degrees of freedom (DoF), consisting of 3 translational and 3 rotational dimensions, with full gravity compensation. It has high-resolution force display capabilities of up to 12 N and position/velocity acquisition with control loop rates of up to 4 kHz. To achieve a fast visual feedback with the lowest possible latency, a 27 *inch* gaming display [Ace16] with 1 ms response time and 144 Hz refresh rate is employed. For the auditory feedback, high-quality headphones with active noise canceling [Bos16] are used. Because the OP and the TOP are located in the same room during the experiments, it is necessary to isolate the human OP acoustically from the test environment with noise cancellation.

2.1.2 Communication network

Communication links between the OP and the TOP provide the exchange of multimodal sensory data from the TOP to the OP and position/velocity signals from the OP to the

TOP. The quality of service (QoS) provided by the network strongly affects the performance of the teleoperation system. In particular, the delay, packet loss and transmission capacity directly influence the system stability and can jeopardize both the task performance of the OP and system transparency. Apparently, the use of a teleoperation system over long-distance wired and wireless networks has challenged engineers working in control, signal processing and networking to design a stable and transparent teleoperation system.

Throughout this thesis, we specifically focus on communication links, such as the internet, that are based on UDP/IPv4 and ethernet protocols. To precisely emulate network behaviors, such as delay, packet loss and transmission rate, a hardware network emulator [App16] is employed in the testbed.

2.1.3 Teleoperator

The TOP is a lightweight robot equipped with multimodal sensors, such as force/torque sensors, cameras, microphones and accelerometers, to capture the physical properties of the remote environment. Additionally, grasping fingers, hands with human anatomy and special tools for specific manipulation tasks can be attached to the robot as an end-effector to interact with the remote environment. As shown in Fig. 2.1, the TOP receives motion commands as position/velocity signals from the HSI, and the local control loop at the TOP side computes the inverse kinematics, which determines the joint positions of the TOP robot for achieving the desired end-effector position.

In this framework, a real teleoperation system was built with the KUKA LWR (Light Weight Robot) arm [KUK] (see TOP robot in Fig. 2.1), which has 7 axes and a maximum payload capacity of 7 kg. The KUKA LWR is a highly sensitive robot, with its integrated sensors in its axes, which makes the robot responsive to command rates of up to 1 kHz. To achieve a precise force sensing at the end-effector, a JR3 multi-axis load cell [JR383], which is also known as a 6-DoF force-torque sensor, was mounted between the last axis of the robot and the manipulation tool.

To perform the bilateral telemanipulation, a control scheme needs to be employed between the OP and the TOP. Several control schemes [HZS01] exist, depending on the manipulation task and features of the haptic device and robot manipulator. The velocity-force control architecture is a commonly used method that directly exchanges the captured velocity of the OP and force feedback from the TOP side. The velocity-force control architecture is illustrated in Fig. 2.1. The haptic device captures the human velocity $\dot{x}_h$, which is transmitted over the network. Similarly, the force feedback from the environment F_e is captured and fed back to the OP over the network. The slave velocity $\dot{x}_s$ and the force feedback F_h displayed to the OP can be the damped versions of the human velocity $\dot{x}_h$ and the environment force feedback F_e depending on the stability requirements of the system.

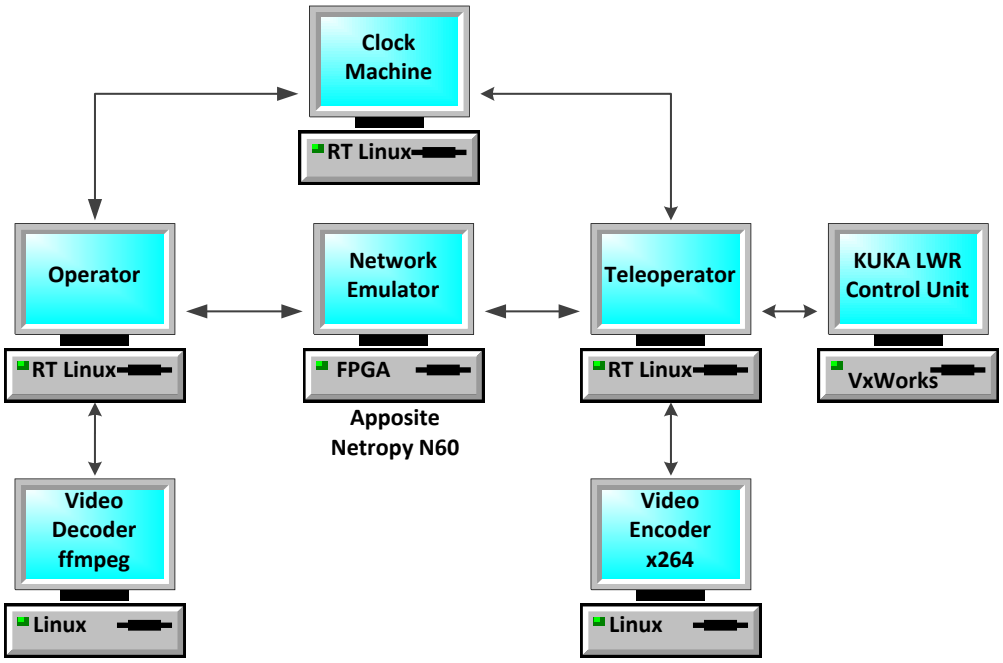


Figure 2.3: Teleoperation system testbed: The physical structure of the teleoperation system is given, and the computers and hardware are interconnected through ethernet-based interfaces.

Fig. 2.3 illustrates the implemented testbed. The OP and TOP computers, running Real-Time (RT) Linux, run the local control loops shown in Fig. 2.1 and are physically separated via ethernet with a hardware network emulator (Apposite Netropy N60). The TOP computer communicates with the KUKA LWR control unit and transmits the computed (X,Y,Z) position of the robot end-effector. The KUKA control unit has its own closed kinematics computation loop for moving the robot joints to the desired end-effector position. The video encoding and decoding are performed on separate machines because the computation load of the video processing interferes with the local control loops at the OP and TOP. Additionally, we use a clock server machine that synchronizes both sides to measure the end-to-end latencies for the evaluation of the system performance.

2.1.4 End-to-end signal latency and its effects on human perception

The effects of latency between a human and a machine was first investigated by Robert Miller in 1968. In his paper [Mil68], he reported that a time delay of 100 *ms* is perceived as instantaneous. In his experiments, he focused on simple events such as keyboard typing and conversational communications. When we consider recent real-time human computer interaction (HCI) applications, these applications may demand delays even lower than 100 *ms*,

depending on the context. In [Che96], Cheshire considered the problem for remote interactive applications over communication networks and reported the sources of latency originating from the hardware and physical limitations of communication. In his white paper, Cheshire concluded that the latency caused by computer hardware and software should be aggressively eliminated. Because it is not possible to beat the speed of light in communications, latency-hiding techniques should be used to give the illusion of interactivity for interactions between geographically distant places. In psychophysics, researchers mostly focus on the human latency perception of visual cues because vision sensing is more dominant in human behavioral responses [PNK76]. In [PWHM14], Potter et al. showed that the human vision system can detect visual cues displayed at 13 ms, which represents an acquisition speed of approximately 75 frames per second. 13 ms is considered as the fastest rate that the human vision system can process, and this is the lowest detectable limit for visual delay from eye to brain. When we consider a human interacting with the environment, the reaction of the body to an event is a complicated task that involves several mental processing stages. First, the sensory perception detects an event; then, this event is passed as an input to the conscious decision process. Once the decision is made, the reaction command is sent to the corresponding part of the body to perform a behavioral response [BW80]. The average reaction time of a human being has been measured to be approximately 200 ms. However, skilled individuals, such as pilots, racers and game players, can have a reaction time as short as 100 ms. It is possible to measure your reaction time with an online application based on a visual event occurring on a screen and a corresponding mouse click, which is considered as the reaction to the visual event [Hum07]. In interactive applications, such as remote game playing and telepresence, the displayed event is delayed due to hardware limitations and communication issues. The sensory perception is the first stage of the reaction process. If the stimuli is delayed, this delay is added to the reaction time of a person, and the increasing stimuli delay has a negative impact on human task performance in an interactive event. A real-time network provider start up called PubNub reported, in one of their articles [Bur15], that a visual delay reaching 75 to 100 ms causes a degradation of the processing ability of a human. At this point, we recognize the lag of the visual input consciously and attempt to adapt our reactions to the slow stimuli. Hence, it is evident that reducing the latency of each modality in our teleoperation system is very important to enhancing the operator's interaction through the system. Therefore, in the following sections, we explain the delay sources of our teleoperation system for end-to-end latency analysis.

Latency of haptic communication

In a real teleoperation system, it is crucial to use a force sensor to acquire the true force signal sensed from the interaction between the robot end-effector and the object. As introduced

above, a JR3 6-DoF force-torque sensor [JR383] and a 6-DoF haptic device, Omega 6, from Force Dimension [For01] are employed in the system at the TOP and OP sides, respectively. In the following, the overall latency on the force signal is given as follows:

$$t^{H}_{delay} = t_{DAC} + t_{network} + t_{display} \tag{2.1}$$

t_{DAC} refers to the delay during force signal filtering by the data acquisition card (DAC). The raw force signal from the sensor is very noisy. The DAC has on-board DSP filters to reduce the noise level. In the JR3 documentation [JR383], it is reported that the group delay of this filter can be approximately computed as follows:

$$t_{DAC} \cong \frac{1}{f_{cutoff}} \tag{2.2}$$

In our teleoperation setup, a filter with a cut-off frequency of 31.25 Hz is sufficient for teleoperation tasks, and in that case, the acquisition delay is approximately 32 ms. $t_{display}$ is the delay that occurs between the computer and the haptic device. From the device API, it is measured as 1 ms. The overall delay on the force feedback can be written as follows:

$$t^{H}_{delay} = t_{network} + 33ms \tag{2.3}$$

In this equation, $t_{network}$ represents the transmission delay of the network.

Latency of audio communication

Interactive applications having hard real-time constraints, such as telepresence and collaborative music, require very-low-latency audio communication to provide a transparent auditory sense. In contrast to video communication systems, audio communication technologies have reached a more mature state. To provide a very-low-delay audio modality for our teleoperation system, the audio codec CELT [VTMM10] is employed. CELT introduces a very low algorithmic delay, depending on the encoding buffer size, of 5 to 22 ms using a full audio bandwidth of 48 kHz. The overall latency observed for the auditory feedback can be derived as follows:

$$t^{A}_{delay} = t_{environment} + t_{acquisition} + t_{encoder} + t_{network} + t_{decoder} + t_{display} \tag{2.4}$$

$t_{environment}$ is the environment delay due to air propagation, which can vary from 5 to 20 ms based on the distance between the microphone and the event location, and $t_{acquisition}$ is the acquisition delay by the soundcard. To read and write the data, we employed an audio I/O library called PortAudio [Por04]. The library API measures $t_{acquisition}$ and $t_{display}$ delays as 12.7 ms each. The encoding and decoding delays, $t_{encoder}$ and $t_{decoder}$, depend on the buffer size setting of the codec and can vary from 5 to 20 ms. The CELT encoder has a constant

bitrate (CBR) mode for real-time streaming and can generate a flat bitstream so that it does not introduce buffer overflow or underflow issues. To achieve the lowest latency of 5 ms for encoding and decoding, the frame buffer size is set to 240 samples at 48 kHz, which results in 200 encoded frames. If the encoder is set to a 64 $kbps$ bitrate, each frame has a size of 40 $bytes$. If we assume that the microphone is placed at the closest point to the event and that the latency is 5 ms, the overall delay can be written as follows:

$$t^{A}_{delay} = 5 + 12.7 + 5 + t_{network} + 5 + 12.7 = t_{network} + 40.4ms \tag{2.5}$$

Latency of video communication

The transmission of a video signal for a teleoperation system has tight delay constraints. Unlike real-time multimedia applications, such as video conferencing, in teleoperation, the OP actively manipulates the remote objects in a closed loop and needs to see his/her manipulation and sense the touch in a remote environment with the lowest possible delay. To achieve this goal, the video communication should provide a good-quality video with very low latency. The glass-to-glass (camera lens-to-display) latency of a video signal can be analyzed as follows:

$$t^{V}_{delay} = t_{camera} + t_{encoder} + t_{network} + t_{decoder} + t_{display} \tag{2.6}$$

where t_{camera} is the image acquisition delay, $t_{encoder}$ is the encoding delay, $t_{network}$ is the transmission delay of the network, $t_{decoder}$ is the decoding delay, and $t_{display}$ is the latency introduced by the monitor. The camera acquisition and display delays are hardware-dependent components, and these delays can be reduced by replacing computer-based systems with custom hardware. For research purposes, commodity hardware, which consists of off-the-shelf computers, and the available camera and display systems, introducing a considerable amount of delay, are employed. This delay can be called the intrinsic delay, $t_{intrinsic}$, which is the sum of acquisition and monitor delays:

$$t_{intrinsic} = t_{camera} + t_{display} = 60ms \tag{2.7}$$

In [BS16], this $t_{intrinsic}$ delay is measured using a blinking LED placed in front of the camera and a photodiode attached to the screen of the video display window. The phase difference between the LED trigger and the photodiode reaction is recorded with a microcontroller [Ard16]. This time difference yields the intrinsic delay, $t_{intrinsic}$. In our teleoperation system, we employed a GigE camera (Allied Vision Mako [All16]) and a 144 Hz gaming display (Acer XB270H [Ace16]), and using the method in [BS16], this delay is measured as 60 ms for 720p high definition (HD) video at 25 fps. Regarding the decoding delay, $t_{decoder}$, the current video decoders are very fast on commodity hardware and can decode a 720p HD video in less than a millisecond. The remaining delay components, $t_{encoder}$ and $t_{network}$, are the

main focus of source coding in video communications. Hardware and software optimizations can be performed to reduce the encoding time, $t_{encoder}$, of a frame. However, controlling the transmission delay, $t_{network}$, is challenging. A video encoder with an accurate bitrate controller is needed to achieve low jitter and optimized delay. In Chapter 4, a detailed description of the rate control problem and its solution will be discussed.

2.2 Haptic communication

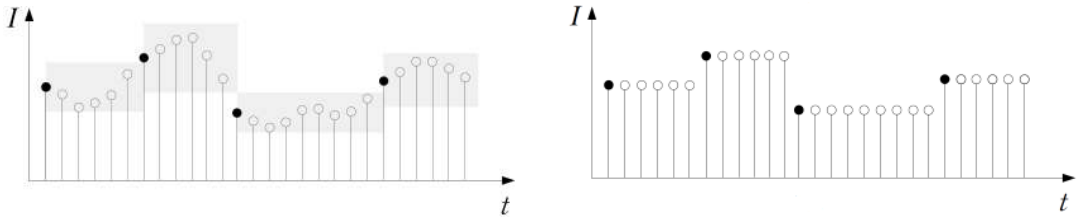
In contrast to audio-video and vibrotactile signals, the transmission of kinesthetic signals, such as force, torque and pressure, is challenging due to the bilateral signal exchange between the OP and the TOP. Apparently, the bidirectional control loop involves the very-low-delay transmission of signals between the OP and the TOP to ensure the stability of a teleoperation system. In this case, it is not possible to apply block-based data compression schemes, which introduce processing group delay during compression [KKHB06]. To minimize the delay, the haptic samples are transmitted immediately as they are captured. On the other hand, the high sampling rate requirement for the control loops is at least 1 kHz for stability reasons [CB94]. The transmission of haptic samples at a high rate, such as 1 kHz, is difficult to realize for complex networks such as when communicating over internet [FI05]. On the other hand, the haptic samples are too small compared to the protocol headers that are added during the transmission. This leads to inefficient usage of network resources. As a solution, sample-based data reduction schemes [HSHB05, HHSB05] have been used. In the following, the perceptually motivated sample-based haptic data reduction scheme originally proposed in [HSHB05] is introduced in detail.

2.2.1 Perceptual deadband coding using Weber's fraction

In psychophysics, it has been shown that the human haptic perception of kinesthetic stimuli can be modeled by a mathematical relationship between the physical intensity of a stimulus and its phenomenologically perceived intensity [Web51]. In 1851, Ernst Weber proved that the magnitude of a difference threshold follows a linear relationship with the stimulus intensity. This relationship has become known as Weber's Law of Just Noticeable Differences (JND):

$$\Delta I = k.I \tag{2.8}$$

where I is the initial stimulus and ΔI is the so-called Difference Threshold (or the JND). The latter indicates the smallest amount of change in the stimulus I that can be detected as often as it cannot be. The constant k (herein called the deadband parameter k) denotes the linear relationship between ΔI and the initial stimulus I. According to Weber's Law, unsubstantial changes in the captured force feedback signal are considered to be unperceivable, and these



(a) Haptic signal with perceptual thresholds (b) Reconstructed haptic signal using ZOH

Figure 2.4: Perceptual deadband coding with zero-order-hold: The height of the gray zones indicates the perceptual deadband regions and is a linear function of the haptic stimulus I. The circles filled with black color represent the haptic samples that violate the applied perceptual thresholds. Thereupon, the black circled samples have to be transmitted to the remote side. At the receiver side, the irregularly sampled signal is interpolated using zero-order-hold reconstruction. This figure is reproduced from [SHK+11] ©2011 IEEE.

haptic samples can be dropped [HSHB05]. When the difference between the recently sent sample and the current signal value violates the human perception thresholds, the current value is sent as a new update. At the receiver side, a basic upsampling method called the zero-order-hold (ZOH) strategy is utilized to interpolate the irregularly received signal samples to the high sampling rate that is required for the local control loops. For the force feedback channel, the principle of deadband-based (DB) data reduction [HSHB05] is shown in Fig. 2.4-a and b. Filled sample circles represent the update samples of the deadband coding scheme. The gray zones illustrate the perception thresholds represented by the perceptual deadband, and the samples inside the zone are interpolated by the ZOH approach. The size of the applied deadband zone is increased directly proportional to the magnitude of the most recently transmitted haptic sample. For haptic perception, this proportionality, k, is constant and has been found to be in the range of 5% to 15%, depending on the type of stimulus and the limb/joint where it is applied [Bur96]. If the haptic signal violates expected human perception thresholds, then a signal update is sent over the network, and the current deadband threshold is also updated with this recent sample. In [HHC+08], Hinterseer et al. performed subjective experiments to compare the quality of experience under different deadband thresholds, and the results indicated that a deadband parameter $k = 10\%$ yields 90% sample rate reduction with satisfactory subjective evaluations. Further extensions have been made to the perceptual haptic data reduction scheme in [HHC+08]. In [KVN+10], Kammerl et al. investigated the perceptual effect of OP velocity and extended the scheme with velocity-adaptive perceptual thresholds. Further extensions have been made to achieve an error-resilient haptic data reduction scheme for lossy communication links, where some packets may become lost during transmission. In [BKS10, BS11, BCS12], the authors showed that haptic packet losses impair teleoperation systems, resulting in instabilities and haptic

artifacts, such as glue and bouncing effects, and they proposed an extension to the perceptual haptic data reduction scheme that triggers additional haptic samples to reduce the haptic artifacts at the receiver side.

2.3 Control architectures with haptic data reduction for time-delayed teleoperation systems

In the haptic data reduction context, another important challenge for teleoperation systems is to design a stable haptic data reduction scheme when there is a considerable amount of bidirectional delay between the TOP and OP. When the slave and master are geographically distant, the communication delay between the TOP and OP plays a critical role in system stability [Law93]. As previously discussed in Chapter 1, passivity-based control architectures and model-mediated teleoperation (MMT) can be employed to provide stable teleoperation under significant latencies. Although MMT can preserve the transparency better than can conventional passivity-based control schemes, it is challenging to model the environment in practice, and new methods are still under development [XCSS16]. In our teleoperation setup, a time-domain passivity-based control architecture is employed to achieve stable teleoperation under considerable communication delay between the TOP and OP [RAP10]. The literature contains passivity-based control architectures combined with haptic data reduction methods [HB07, VKHS09]. In these approaches, authors have attempted to combine the wave-variables transformation [AS89b] with deadband-based haptic data reduction. However, these methods assume that the communication delay is not rapidly changing over time and that the delay is a known parameter in the system. On the other hand, because the wave-variables approach transforms the haptic signals into the wave domain, it is not straightforward to apply deadband thresholds using the perceptual boundaries. In the following, we introduce a perceptual haptic data reduction scheme integrated into a time-domain passivity control architecture [RAP10], which is proposed in [XCSS15].

2.3.1 Time-domain passivity-based haptic data reduction

In our teleoperation system, we employed a time-domain passivity-based control architecture [RAP10] and a perceptual haptic data reduction scheme [HHC+08]. In Fig. 2.5, we illustrate the TDPA-based control architecture and haptic data reduction processing blocks. The parameters $E^m_{in}(t)$, $E^s_{in}(t)$ and $E^m_{out}(t)$, $E^s_{out}(t)$ denote the incoming and outgoing energy flows at the master and slave sides, respectively. The following equations are used to determine the energy flows.

$$E^m_{in}(t) = \begin{cases} E^m_{in}(t-1) + \Delta E^m(k), & if \quad \Delta E^m(t) > 0 \\ E^m_{in}(t-1), & else \end{cases} \tag{2.9}$$

$$E^m_{out}(t) = \begin{cases} E^m_{out}(t-1) - \Delta E^m(t), & if \quad \Delta E^m(t) < 0 \\ E^m_{out}(t-1), & else \end{cases} \tag{2.10}$$

$$E^s_{in}(t) = \begin{cases} E^s_{in}(t-1) + \Delta E^s(t), & if \quad \Delta E^s(t) > 0 \\ E^s_{in}(t-1), & else \end{cases} \tag{2.11}$$

$$E^s_{out}(t) = \begin{cases} E^s_{out}(t-1) - \Delta E^s(t), & if \quad \Delta E^s(t) < 0 \\ E^s_{out}(t-1), & else \end{cases} \tag{2.12}$$

where $\Delta E^m(t) = v_m(t) f_m(t) \Delta T$ and $\Delta E^s(t) = v_s(t) f_s(t) \Delta T$ are the energy changes at the master and slave sides, respectively; t denotes the sampling instant; and ΔT is the sampling period. f_m and v_m are force and velocity signals at the master side, and f_s and v_s are force and velocity signals at the slave side. Because the energy flows on the master and slave sides are positive and monotonically increasing, the passivity condition [RAP10] is expressed as follows:

$$E^m_{in}(t) + E^s_{in}(t) \geq E^m_{out}(t) + E^s_{out}(t) \tag{2.13}$$

In [RAP10], a sufficient and conservative condition that satisfies the above passivity constraint is given as follows:

$$E^m_{in}(t) \geq E^s_{out}(t) \quad \text{and} \quad E^s_{in}(t) \geq E^m_{out}(t) \tag{2.14}$$

It is important to note that $E^m_{in}(t)$ and $E^m_{out}(t)$ are determined at the master side and that $E^s_{in}(t)$ and $E^s_{out}(t)$ are determined at the slave side. To observe the system passivity using Eq. 2.14, $E^m_{in}(t)$ and $E^s_{in}(t)$ need to be exchanged over the network. However, as seen from Fig. 2.5, the communication network delays the transmitted energy information with instant delays of $T_1(t)$ from master to slave and $T_2(t)$ from slave to master. Due to the monotonically increasing input/output energies, it is still sufficient to satisfy the passivity constraint given in Eq. 2.13 using the modified version of Eq. 2.14, including the time-delay shifts, as follows:

$$E^m_{in}(t - T_1(t)) \geq E^s_{out}(t) \quad \text{and} \quad E^s_{in}(t - T_2(t)) \geq E^m_{out}(t) \tag{2.15}$$

If the passivity condition given in Eq. 2.15 is violated, passivity control (PC) is applied to the received velocity on the slave side and to the received force on the master side. In this case, the adaptive dampers α and β are enabled on the master and slave sides, respectively.

These dampers dissipate the output energy to preserve the system passivity. If we consider the dissipated energies, E^m_{PC} and E^s_{PC}, due to the adaptive dampers, α and β, on the master and slave sides, the passivity condition given in Eq. 2.15 is formulated as its final version in the following:

$$\begin{aligned} W_m(t) &= E^s_{in}(t - T_2(t)) - E^m_{out}(t) + E^m_{PC}(t-1) \geq 0 \\ W_s(t) &= E^m_{in}(t - T_1(t)) - E^s_{out}(t) + E^s_{PC}(t-1) \geq 0 \end{aligned} \tag{2.16}$$

Based on the final passivity condition given in Eq. 2.16, the adaptive dampers, α and β, are given as follows:

$$\alpha(t) = \begin{cases} 0, & if \quad W_m(t) > 0 \\ -\frac{W_m(t)}{\Delta T v^2_{mc}(t)}, & else \quad if \quad |v_{mc}(t)| > 0 \end{cases} \tag{2.17}$$

$$\beta(t) = \begin{cases} 0, & if \quad W_s(t) > 0 \\ -\frac{W_s(t)}{\Delta T f^2_s(t)}, & else \quad if \quad |f_s(t)| > 0 \end{cases} \tag{2.18}$$

Finally, the dissipated energies are updated for the next iteration as follows:

$$\begin{aligned} E^m_{PC}(t) &= \Delta T \sum_{j=0}^{t} \alpha(j) v^2_{mc}(j) \\ E^s_{PC}(t) &= \Delta T \sum_{j=0}^{t} \beta(j) f^2_s(j) \end{aligned} \tag{2.19}$$

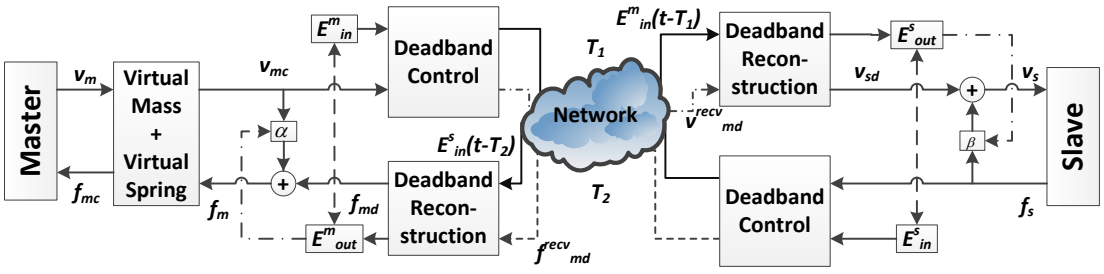


Figure 2.5: TDPA-based haptic data reduction approach: The control architecture of the developed teleoperation system. Perceptual deadband-based data reduction with ZOH is employed inside the data reduction blocks. The passivity observers and controllers ensure the stability of the communication network and data reduction blocks. This figure is reproduced from [XCSS15] ©2015 IEEE.

As illustrated in Fig. 2.5, passivity observers (POs) calculate the input and output energy on both the master and slave sides. Concurrently, the POs check the passivity condition based on the locally evaluated output energy and received input energy from the remote side. If the passivity constraints are matched, the received velocity or force signal is directly displayed. If the passivity constraints are not satisfied, the PCs are enabled, as shown in Eq. 2.20, with the adaptive dampers, α and β, given in Eq. 2.17 and Eq. 2.18, to dissipate the output energy to ensure system passivity:

$$\begin{aligned} v_s(t) &= v_{sd}(t) + \beta(t) f_s(t) \\ f_m(t) &= f_{md}(t) + \alpha(t) v_{mc}(t) \end{aligned} \tag{2.20}$$

Furthermore, as seen in Fig. 2.5, the virtual mass and spring model is applied as a passive low-pass filter to the velocity and force signals (we refer the reader to [RAP10] for more details).

In [XCSS15], we extended the passivity control architecture, including the perceptual haptic data reduction blocks, as observed in Fig. 2.5. The blocks called "Deadband control" on both the master and slave sides apply the data reduction to the force and velocity samples using the perceptual deadband data reduction approach discussed in Section 2.2.1. When there are no updates, the blocks called "Deadband Reconstruction" apply ZOH reconstruction. In the data reduction case, the energy changes are detected by the POs as follows:

$$\Delta E^m(t) = \begin{cases} v_{mc}(t) f_{md}^{recv}(t) \Delta T, & if \quad signal \quad received \\ v_{mc}(t) f_{md}(t^*) \Delta T & else \end{cases} \tag{2.21}$$

$$\Delta E^s(t) = \begin{cases} v_{sd}^{recv}(t) f_s(t) \Delta T, & if \quad signal \quad received \\ v_{sd}(t^*) f_s(t) \Delta T & else \end{cases} \tag{2.22}$$

where $t^* < t$ is the time instant of the most recently received signal update, $v_{sd}(t^*)$ and $f_{md}(t^*)$ are the most recently received velocity and force signals, and $v_{sd}^{recv}(t)$ and $f_{md}^{recv}(t)$ denote the currently received velocity and force signals at the slave and master sides, respectively. According to Eqs. 2.9 - 2.12, the input and output energies on the master and slave sides are computed based on the signs of the energy change. It is important to note that the ZOH reconstruction scheme is non-passive [VKHS09], and the authors in [VKHS09] had to use the following passive deadband reconstruction:

$$f(t) = f(t^*) - sign(v_s) \Delta_f \tag{2.23}$$

where $t^* < t$ is the time instant of the most recently received signal, $f(k^*)$ is the most recently received force signal, and Δ_f is the DB zone defined by the most recently received signal $f(t^*)$

and the DB parameter k as follows:

$$\Delta_f = k \cdot f(t^*) \tag{2.24}$$

The passive ZOH reconstruction damps the incoming energies compared to simple ZOH. However, in contrast to the simple ZOH reconstruction, the passive ZOH approach can impose a higher signal error between the real and reconstructed signal, which can lead to higher signal jumps when a new sample is received. The TDPA approach provides an advantage for placing the DB control and the reconstruction blocks at the network input ports such that we can consider the DB blocks and network as a combined, non-passive 2-port network. Therefore, the TDPA scheme ensures the stability for this combined non-passive effect.

In [XCSS15], subjective experiments were conducted to compare the performance between the perceptual haptic data reduction schemes combined with TDPA-based [XCSS15] and WV-based [VKHS09] control architectures. During the tests, the round-trip time was set to a 100 ms constant delay, and the tested DB parameters were 0%, 2%, 5%, 8%, 10% and 15%. During the experiment, subjects performed 6 trials under 6 randomly selected DB parameters. In every trial, the subjects were asked to compare the TDPA-based and WV-based data reduction schemes with a common reference, and they determined which scheme was closer to the reference and which scheme was preferable. In Table 2.1, the percentages of the preferences are given; we see that the subjects found that TDPA-based perceptual haptic data reduction had a closer impedance to the common reference and consequently preferred it as the best scheme.

Table 2.1: Subjective impedance tests and system preference [XCSS15].

Subjective impedance similarity						
Deadband (%)	0	2	5	8	10	15
TDPA-based	73%	80%	73%	80%	93%	100%
WV-based	27%	20%	27%	20%	7%	0%
Scheme preference						
Deadband (%)	0	2	5	8	10	15
TDPA-based	73%	87%	80%	80%	93%	93%
WV-based	27%	13%	20%	20%	7%	7%

Because the TDPA-based haptic data reduction scheme [XCSS15] was considered to be subjectively the preferred approach, we employ it in our teleoperation system.

2.4 Related work

In the literature, research on transmission schemes for systems involving haptic feedback can be categorized into two areas: approaches based on the transport layer and approaches based on the application layer. In the following, we report the related work performed in this context.

Transport layer approaches

1. In [UY04], Uchimara et al. presented a bilateral teleoperation system and focused on reducing the delay occurring between the application and physical layers due to the operating system scheduling. In their approach, they placed an interrupt handler between the data-link and physical layers to directly communicate with the real-time control task. The protocol forwards non-real-time packets to the upper layers of the OSI model, namely, the network, transport, session, presentation and application layers, in the respective order. On the other hand, the real-time traffic is directly passed to the real-time process. In their experiments, they showed that the side traffic coming along side of the teleoperation session interrupts the real-time control timings by producing a delay-jitter longer than the loop period of 1 ms. However, when the real-time network structure and protocol proposed by the authors is enabled, the delay-jitter of the control signals is reduced to 0.5 ms, which is acceptable for bilateral teleoperation systems.

2. In [PWZ05], Ping et al. replaced the existing UDP and TCP protocols with a new transport layer protocol called the Interactive Real-Time Protocol (IRTP), which provides the advantages of both UDP and TCP for internet robot control systems. IRTP can be considered as a hybrid protocol that is the combination of existing UDP and TCP. IRTP classifies the streams as unreliable or reliable internally, and it employs a window-based flow control to predict the bandwidth and fully utilize the capacity.

3. In [CML+05], the authors implemented a transport layer QoS management scheme on top of overlay networks for bilateral teleoperation systems involving multimodal interaction. They applied task dexterity detection to identify the priority of each media stream, and the available bandwidth was allocated based on the weighted priorities. Furthermore, they applied traffic shapers to ensure the allocated rate to the corresponding stream.

4. In [CMZX05], Cen et al. worked on a transport layer service for bilateral control systems running over relayed wireless networks. The proposed scheme determines the optimum paths for transmitting the teleoperation streams by measuring the QoS parameter of

each path to the destination. The scheme dynamically generates a transport plan based on the media stream QoS requirements and forwards them over multiple paths.

Application layer approaches

1. In [CSKR07, CHK+09], Cha et al. proposed a haptic broadcasting system for passive interactions using a multimedia framework based on MPEG-4 BIFS (Binary format for scenes). The haptic interaction media are multiplexed into MPEG-4 scene description and aligned spatio-temporarily with the audio-visual data. The scheme is basically an extended container format that enables haptically enhanced broadcasting for video on-demand systems.

2. In [ITN11, KNT15], the authors proposed QoE enhancement schemes with intra-stream synchronization for bidirectional haptic interactions for a teleteaching tool. In their application, the audio, video and haptic streams flow in both directions between the instructor and manipulator nodes. The application was tested over a 10 $Mbps$ CBR link loaded by the media streams having average bitrates of 5.8 to 6.2 $Mbps$, and they applied application layer transmission schemes with media adaptive playout buffering, skipping and buffering haptic samples. According to the user experience evaluation, the media adaptive buffering with haptic sample skipping performed the best subjectively. However, this work lacks haptic data reduction methods because the heavy load of the packet traffic can be reduced by the perceptual haptic data reduction approach, which also provides a satisfactory user experience.

3. In [ECES11], Eid et al. conducted a comprehensive study on haptic-audio-visual data communication protocols, and they proposed a scheme for application layer statistical multiplexing for a multimodal tele-immersion system involving haptic interaction. In their approach, the scheme allocates resources based on a statistical switching mechanism between modalities until the delay constraint of the signal is close to being violated. Furthermore, they reported that the QoS requirements are not always satisfied. Therefore, the scheme needs to employ delay and jitter compensation modules to ensure stability if it is applied for a bilateral teleoperation system. On the other hand, they mentioned that advanced data reduction and rate control techniques need to be employed for audio, video and haptic signals to efficiently use the available transmission resources.

4. In [YTY13, YYYK14], the authors addressed the transmission capacity problem when force feedback and video frames are transmitted together over CBR links having a transmission capacity of $4 - 10$ $Mbps$, and they employed an end-to-end flow controller to adapt the packet rate and bitrate of the visual-haptic streams. They showed that

video frames generated by a JPEG [ITU93] encoder block the haptic packets and cause additional queueing delays. The authors applied adaptive selection for the bitrate and frame rate of the video stream and packet rate of haptic samples based on a transmission rate estimation scheme and a queueing observer, which can be considered as a congestion detector. Although this approach has similarities with the methods that are being discussed in this thesis, the scheme does not employ state-of-the-art haptic data rate reduction techniques, and the visual communication system applies primitive coding approaches, such as JPEG and frame rate reduction down to 3 fps, which causes high delay and low visual quality according to the human vision system.

2.5 Chapter summary

In this chapter, we first introduced the building blocks of a haptic teleoperation system using off-the-shelf consumer robotics hardware and open-source software available on the web. Second, we addressed the haptic communication issues caused by the high data rate and time delay. Then, the perceptual haptic data reduction scheme and the time-domain passivity control approach, which are the core blocks of the teleoperation system studied in this thesis, were described. The following items are the most important aspects to highlight:

- The hardware setup of the teleoperation system and its implementation are explained in detail.
- The end-to-end latency of each modality is measured in this thesis, and we find that most of the delays come from signal acquisition and display.
- We employ a state-of-the-art perceptual haptic data reduction scheme together with a time-domain passivity controller for a real teleoperation setup.
- Similar works in the literature are summarized. Their achievements and weaknesses are discussed.

Chapter 3

Considered Teleoperation Scenario and Motivation

As shown in Fig. 3.1, the TOP senses the remote environment and sends the multimodal information to the human OP over the communication network. The QoS provided by the network strongly influences the performance of the teleoperation system. In particular, the delay, packet loss and transmission capacity negatively affect the system stability and jeopardize the task performance of the OP and system transparency. Apparently, the usage of a teleoperation system over long-distance wired and wireless networks represents a challenge to the design of a reliable and stable teleoperation system. In particular, if the network is shared with other streams, such as the internet, the transmission rate may fluctuate over time due to the unknown side traffic along the communication path. As observed in Fig. 3.1, the

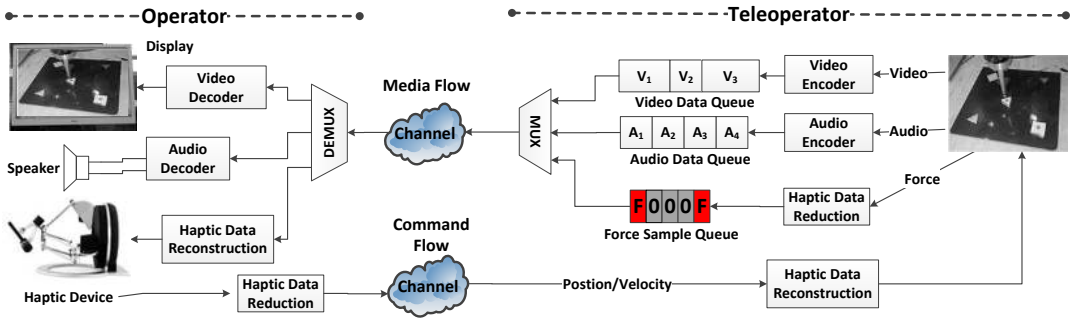


Figure 3.1: Schematic overview of a multimodal teleoperation system: The multimodal signals are encoded and multiplexed into a single stream to be transmitted over the communication channel. The demultiplexer is responsible for splitting the signals and forwarding them to the corresponding display.

different modalities need to be multiplexed into a single stream according to a priority-based transmission rate sharing strategy to ensure the efficient utilization of the network capacity. In this thesis, we focus on the multiplexing of audio-video and force feedback signals for teleoperation sessions running over networks having low transmission rate capacity. The next section introduces the delay effect under low transmission rate conditions, and we build our proposed solution step by step in subsequent sections.

3.1 Problem statement

The inclusion of the audio and video streams in haptic communication produces another networking challenge to be solved. If the transmission capacity of the network is low and limited, the transmission rate budget should be fairly distributed among the audio-video and haptic streams. Compared to the auditory and visual stimuli, haptic feedback is more sensitive to latency. Hence, the transmission of the haptic packets should be highly prioritized during the communication. The main focus of this thesis is on solving the transmission resource-sharing problem for multimodal streams over communication links with limited transmission rates. Such conditions may exist in earth-to-space communication for on-orbit teleservicing [GOU+12], wide area networks connected via satellite-internet connections [PHP+07] and troposcatter links that are used as point-to-point wireless links [DA15].

In Fig. 3.2, we illustrate the resource sharing problem between video and haptic packets. In this example, we consider a 1 $Mbps$ CBR link between the TOP and OP. Assuming that the scheduling is done based on the first-come first-serve (FCFS) discipline, the packet arrival times at the TOP side and in the serving scenario are as follows:

1. At $t_{V1} = 0$ ms, a packet containing the bit stream of video frame V_1 arrives, and its transmission is scheduled immediately, as the channel is assumed to be idle at time instant 0 ms. The channel service time for the video packet V_1 is 32 ms because the packet size is 4000 $bytes$ and because the transmission rate is 1 $Mbps$.

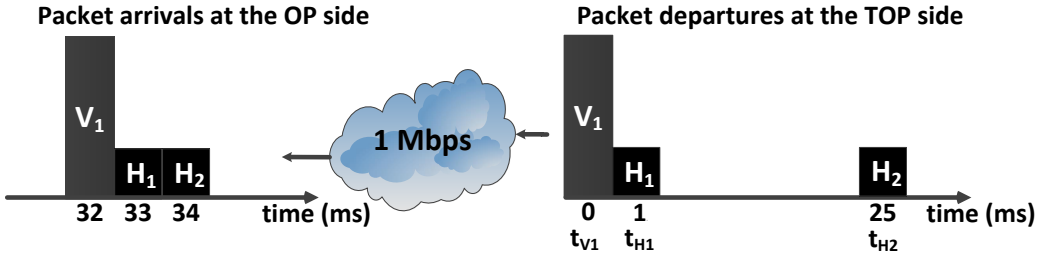


Figure 3.2: Transmission hold up for haptic information caused by a large video segment.

2. Slightly after video packet V_1, haptic packet H_1 is ready for transmission at $t_{H1} = 1$ ms, and it is scheduled to be transmitted after video packet V_1.

3. At time $t_{H2} = 25$ ms, another haptic packet, H_2, is triggered for transmission, and it is scheduled to be transmitted after the preceding haptic packet, H_1.

4. For simple analysis, we assume in this example that the service time of a haptic packet is 1 ms and we ignore the transmission medium headers.

In Fig. 3.2 on the left, the packet arrival times at the OP side are given for each packet type. The haptic packets encounter blocking delays due to the previously scheduled large video packet V_1. Haptic packets H_1 and H_2 are delayed by 32 and 9 ms, respectively. The most critical problem from a teleoperation perspective is the varying delay (jitter) of the haptic packets because the jitter distorts the regular display instants of the haptic samples, which may lead to misperception of the remote environment and may also cause instability problems. In this example, the interarrival distance between consecutive haptic packets is $t_{H2} - t_{H1} = 25 - 1 = 24$ ms, and this distance becomes 1 ms at the OP side (see arrivals at the OP side in Fig. 3.2). Because we expect variable video packet sizes and irregularly triggered haptic samples, this jitter is unavoidable for the haptic packets, which may lead to instability problems for teleoperation systems.

3.2 Scheduling the transmission of video and haptic signals

In a teleoperation session, to avoid large blocking delays for the haptic packets as shown in Fig. 3.2, the haptic packets need to be highly prioritized, and their immediate transmission must be ensured as they become available at the multiplexer. The most suitable queuing discipline for such a scenario is preemptive and resume scheduling, wherein a packet with a high priority interrupts the transmission of a lower priority packet, and the remaining transmission of the lower priority packet is resumed after transmission of the high-priority packet. To realize this preemptive resume functionality, in [Wal05], the authors divided the server capacity into equal discrete-time serving slots, whereby the service time for a particular packet can be determined by the number of used slots. For instance, this type of modeling is applicable for analyzing the performance of Asynchronous Transfer Mode (ATM) switches because ATM networks communicate using fixed-length (53 bytes) cells, and the transmission time of a cell is constant. Using the same perspective, the transmission rate for a teleoperation system can be sampled discretely at 1 kHz, which is the sampling rate of the haptic signal. With this approach, we have 1 ms accuracy to control the transmission service time of each packet. For example, in Fig. 3.2, if we constrain the fixed-length transmission slots sampled with a period of 1 ms, each slot occupies 125 *bytes*, similar to the 53 *bytes* of ATM cells. In this case, video

packet V_1 can be transmitted in 32 slots. With this scheme, the important haptic packet H_1, which is ready for transmission at $t_{H1} = 1\ ms$, is preempted. The transmission of the video packet is paused, and the haptic packet, H_1, immediately receives the opportunity to use the channel. Once the haptic packet is transmitted, the video transmission is resumed at the point at which it was interrupted. According to the new scheduling, the total delay for the haptic packets shown in Fig. 3.2 drops to 1 ms, and the total delay of the video packet is increased from 32 to 34 ms because it is held for 2 ms during haptic packet transmissions. Fig. 3.2 describes only a specific example for illustration purposes. Nonetheless, the size of a video frame is variable; even when the bitrate is controlled by an algorithm, and the haptic packet arrivals occur irregularly. In the following, we consider the problem stochastically to better understand the delay effects mentioned above and approach the problem with applied queuing theory. We assume that the sample arrivals and service time of the channel are stationary processes with independent and identically distributed (iid) samples. Let the arrival random processes of the video and haptic streams be A_H and A_V, respectively, and their corresponding service time random processes be S_H and S_V. Haptic arrivals A_H are modeled as a Poisson random process with mean $\mu_{A_H} = 20\ packets/second$, and video arrivals are deterministic ($\mu_{A_V} = 25\ packets/second$ and $\sigma^2_{A_V} = 0$) because the frame rate is fixed at 25 fps. The haptic packets can be served very quickly; therefore, the service time for haptic packets can be considered as deterministic ($\mu_{S_H} = 1\ ms$ and $\sigma^2_{S_H} = 0$). The service time of video frames depends on the video frame size and the transmission rate according to the following linear relations.

$$\mu_V = E[S_V] = \frac{E[X]}{C} \tag{3.1}$$

$$\sigma^2_{S_V} = Var[S_V] = E[S_V^2] - E[S_V]^2 = \frac{E[X^2]}{C^2} - \frac{E[X]^2}{C^2} = \frac{Var[X]}{C^2} \tag{3.2}$$

where X is the random process generating the video frame size in $bytes$ and C is the constant transmission rate. The frame size statistics of the video stream with a bitrate of 800 $kbps$ and a frame rate of 25 fps are assumed to follow a Gaussian distribution with $\mu_X = 4000\ bytes/frame$ and $\sigma_X = 1500\ bytes/frame$. Using the given parameters, we perform queuing simulations under the following conditions and report the delay-jitter results in Table 3.1.

1. The FCFS principle is used to schedule the packets.
2. Haptic samples are highly prioritized with preemptive and resume scheduling.
3. Items (1) and (2) are performed again if we use a precise video bitrate control algorithm, which generates a flat constant bitrate video stream, where $\sigma_X = 0$.

As observed in Table 3.1, the minimum latency conditions can be achieved using preemptive resume scheduling with an accurate video bitrate controller. The usage of an efficient

Table 3.1: FCFS and preemptive resume scheduling comparison: The table presents the mean and standard deviation (jitter) of the delays for haptic samples and video frames.

Scheduling	**Without video BR control**				**With video BR control**			
	$\mu_{D_{Haptic}}$	$\sigma_{D_{Haptic}}$	$\mu_{D_{Video}}$	$\sigma_{D_{Video}}$	$\mu_{D_{Haptic}}$	$\sigma_{D_{Haptic}}$	$\mu_{D_{Video}}$	$\sigma_{D_{Video}}$
FCFS	$18ms$	$15ms$	$35ms$	$14ms$	$13ms$	$10ms$	$32ms$	$0ms$
Preemptive	$1ms$	$0ms$	$38ms$	$16ms$	**1ms**	**0ms**	**33ms**	**1ms**

bitrate controller significantly reduces the queuing delay and jitter of the visual feedback. The average queuing delay reduction can also be shown theoretically using Pollaczek-Khintchine's formulas with Kingmann's approximations for the General arrival rate distribution/General service time distribution/single server (G/G/1) queue model [Ive15]. The average waiting time (W_q) of a frame in the queue can be computed as follows:

$$W_q = \frac{\rho(\alpha_{a_v}^2 + \alpha_{s_v}^2)}{2(1-\rho)} E[S_V] \tag{3.3}$$

where $\alpha_{a_v}^2 = \frac{Var[A_V]}{E[A_V]^2}$ and $\alpha_{s_v}^2 = \frac{Var[S_V]}{E[S_V]^2}$ are the squared coefficients of variation for the arrival of frames and the service time, respectively, and $\rho = \frac{BR}{C}$ is the link utilization obtained from the average video bitrate BR and communication capacity C. Because the video arrival rate is deterministic, its squared coefficient of variation is zero, $\alpha_{a_v}^2 = 0$, which simplifies the problem to the D/G/1 queue model (D refers to deterministic arrivals). By substituting the relations above, the expressions are simplified to the following:

$$W_q = \frac{\rho}{2C(1-\rho)} \frac{Var[X]}{E[X]} \tag{3.4}$$

As observed from the final expressions 3.2 and 3.4, the service time deviation (σ_{S_V}, jitter delay) and waiting time (W_q) in the queue depend on the variance of the video frame sizes ($\sigma_X^2 = Var[X]$). If the variance of the video frame sizes is reduced to close to zero ($\sigma_X^2 \to 0$), the variance of the service time and the mean waiting time in the queue are also reduced to close to zero ($\sigma_{S_V}^2 \to 0$ and $W_q \to 0$). With this approach, the D/G/1 queuing problem can be reduced to a D/D/1 (deterministic arrival/deterministic service time/single server) problem, which allows us to control and constrain the video transmission delay if the server capacity C is known or estimated correctly. To achieve this state, we showed that the usage of an accurate frame rate controller during video encoding is necessary to reduce the variance of the frame sizes ($\sigma_X^2 \to 0$). In Chapter 4, we introduce the rate control scheme developed for the teleoperation system in detail, and the low-delay transmission of video frames using a single-frame delay constraint is described in Chapter 5.

3.3 Multiplexing audio, video and haptic signals for teleoperation systems

As we have shown theoretically in Section 3.2, the preemptive and resume strategy can stream the haptic samples with high priority while introducing slightly more video delay than under the FCFS strategy. However, this has certain drawbacks in practice. To achieve 1 ms delay accuracy, the packets are constrained to small, fixed-size slots, and consequently, we reach a high packet rate of 1000 $packets/second$. This causes an inefficient usage of network resources due to header usage for the transport protocols. For instance, almost 34% of the transmission rate would be used for the headers if UDP/IPv4 was used over a CBR 1 $Mbps$ ethernet medium because, for every packet, the protocols add 42 $bytes$ of header data to stream the packet. Additionally, the high packet rate may quickly saturate the buffers of transmission media in the network, and some of the packets may be dropped consequently. In the following, we introduce a detailed network delay model for shared data networks, such as the internet, and propose a practical approach for multiplexing multimodal signals.

3.3.1 Delay model for data networks

Having an appropriate network delay model strongly influences the design and performance of communication protocols to predict the delay that is required to transmit a packet or a specific signal segment (such as video, audio frames and haptic samples) from source to destination. Therefore, it is necessary to derive a mathematical model of the communication delay and understand its effects on the transmitted data. In Section 3.2, we studied the transmission capacity problem using applied queuing theory from a high-level perspective. Relying on the previous derivations, in this section, we provide a methodological approach to predict the communication delay in a data network.

In Fig. 3.3, a basic communication scenario between two nodes is given. In this example, the packet streams P and X arrive at node N_1 from two paths, are scheduled for transmission and are forwarded to node N_2. In [BG92], the authors showed that the delay on a packet from one node to another can be modeled as the sum of the following four components:

$$t_{network} = t_{proc} + t_{queue} + t_{channel} + t_{prop} \tag{3.5}$$

- *Processing delay*, t_{proc}, is the time difference between the arrival of the packet at the head of the node and the assignment of the packet to an outgoing path to the desired destination node. This packet processing is negligible and takes a few microseconds using today's networking hardware unless a complicated encryption scheme is applied or the packet's content is examined for security issues.

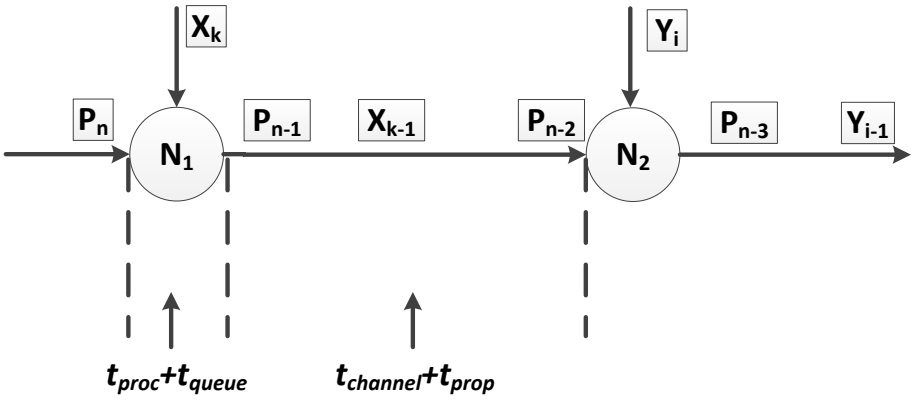


Figure 3.3: The delay model of a packet between two nodes.

- *Queuing delay*, t_{queue}, is the duration that the packet waits in the outgoing queue until its transmission starts. This can also be called multiplexing delay, which is introduced by the transmission scheduling strategy for the incoming packets. The transmission capacity C between N_1 and N_2 is either known or predicted by N_1. Therefore, the multiplexer performs the rate allocation by scheduling the incoming streams P and X.

- *Channel delay*, $t_{channel}$, is the difference between the departure time instants of the last and first bits of a packet from node N_1. This delay is related to the capacity (C) of the link between two nodes in terms of $bits/second$ and the length (L) of the packet in $bits$. Therefore, the channel delay is computed as $t_{channel} = \frac{L}{C}$.

- *Propagation delay*, t_{prop}, is the travel time for the first bit of the packet from the sender N_1 to the receiver N_2. The propagation delay is often confused with the channel delay. However, it is caused by the propagation speed (S), which depends on the physical characteristics of the communication medium. For example, a link composed of copper wire has a speed of 2×10^8 m/s, and a wireless link can reach a speed of 3×10^8 m/s. The propagation delay, t_{prop}, is proportional to the physical distance (D) between transmitting (N_1) and receiving (N_2) nodes and is computed as $t_{prop} = \frac{D}{S}$. If the distance between N_1 and N_2 is quite long, a substantial propagation delay is observed for either a low-capacity satellite link or a very-high capacity fiber optic link.

In Fig. 3.3, the delay between two nodes of a network is illustrated. Regardless of the topology of the network, the delays between nodes in a point-to-point communication path can be accumulated to approximately compute the overall delay of a packet. Queueing and channel delays can be considered as a combined transmission delay ($t_{trans} = t_{queue} + t_{channel}$),

which represents the delay due to the available transmission rate of the communication path. The propagation delay, t_{prop}, is often a constant delay, and this delay can be predicted by sending small packets, which are exposed to negligible queueing and transmission delays over the communication path. Let TR be the available transmission rate in $bits/second$, and let L_n be the length of packet P_n in $bits$. Then, the transmission delay of packet P_n is computed as follows:

$$t_{trans}^{P_n} = \frac{L_n}{TR} \tag{3.6}$$

Therefore, for any point-to-point communication path, the overall packet delivery delay can be represented by a constant propagation delay and its transmission delay as follows:

$$t_{network}^{P_n} = t_{prop} + \frac{L_n}{TR} \tag{3.7}$$

However, Eq. 3.7 is valid only if the transmission of the preceding packet P_{n-1} has already been completed before the arrival of P_n at node N_1. In other words, the interarrival time of the packets should be greater than the service time (transmission time) of the packets to verify this condition. Otherwise, the successive packets experience additional queuing delay called the waiting time (W_q), which was previously defined in Eq. 3.3. In this case, the packet P_n has to wait for the transmission of its preceding packets, which are buffered in a queue at node N_1. Let j be the number of packets in the transmission queue of node N_1; then, the expected delivery time of packet P_n can be derived as follows:

$$t_{network}^{P_n} = t_{prop} + \frac{\sum_{i=n-j}^{n-1} L_i + L_n}{TR} \tag{3.8}$$

where the waiting time in queue for packet P_n is $W_q^{P_n} = \frac{\sum_{i=n-j}^{n-1} L_i}{TR}$. Reconsidering Kingmann's approximation in Eqs. 3.3 and 3.4, it has been shown that a waiting delay close to zero ($W_q \to 0$) is achievable if the system ensures deterministic packet arrivals and lengths simultaneously. Assuming that TR is known or predicted correctly at the sender side and does not fluctuate very rapidly, an application-layer multiplexer that adjusts the packet lengths and the interarrival distance between consecutive packets can be designed. Hence, with Eq. 3.7, the delay of the packets can be controlled according to the needs of the application.

3.3.2 Proposed multiplexing scheme

The multiplexing scheme has been initially proposed in [CCX+14] for teleoperation systems communicating over CBR links. The proposed approach uniformly divides the available transmission rate TR into 1 ms discrete resource buckets and controls the size of the transmitted video packets as a function of the irregular haptic transmission events generated by the perceptual haptic data reduction approach. Because the haptic samples arrive irregularly, the multiplexing scheme [CCX+14] uses a buffer with a fixed size for the force samples to observe

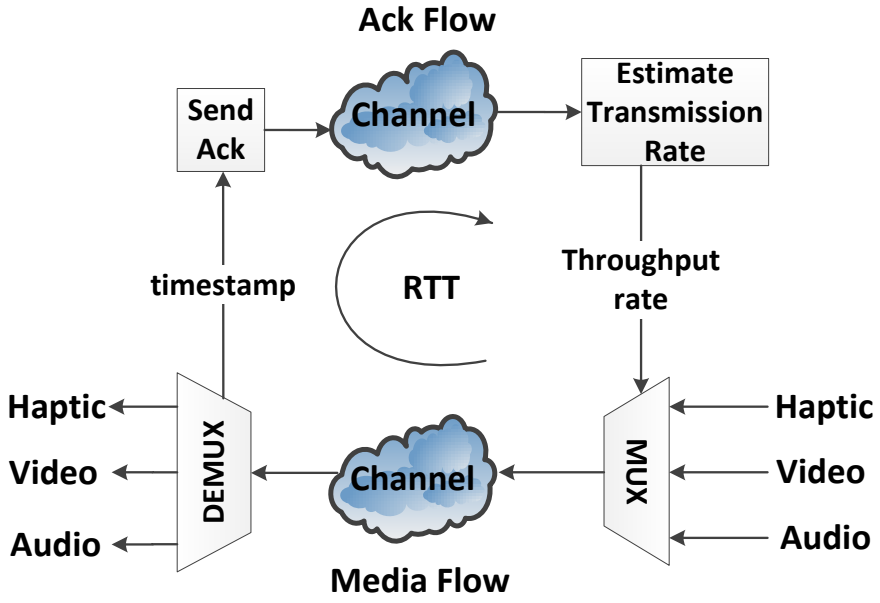


Figure 3.4: A closed-loop multiplexing system: The scheme can also be considered as a queuing system with feedback to determine the uncertain service time. In this illustration, the bottleneck is the channel transmitting the multiplexed media flow. The acknowledgment flow bitrate is very small. Therefore, it is assumed that a sufficient data rate exists in the ack channel for sending the packets quickly. Round trip time (RTT) represents the two-way propagation delay, $RTT = t_{prop}^{forward} + t_{prop}^{backward}$.

the transmit and non-transmit states of the haptic data reduction scheme. With the multiplexing buffer, the scheme is able to foresee the upcoming transmission slots and determines the preemption and resume times at the application layer. Additionally, if consecutive non-transmit slots exist in the multiplexing buffer, the transmission slots can be merged to reduce the packet rate and network header usage. Among the related studies, [CML+05, PWZ05] considered the available network transmission rate at the transport layer and applied traffic shaping and packet dropping to the streams. However, the available transmission rate could also be estimated at the application layer to adapt to the source encoders. In this thesis, we assume that the network path is unknown and that there is no special priority-based QoS management applied by the relaying routers, and the available transmission rate is not rapidly fluctuating. Congestion control of this framework is performed fully at the application layer, and the bitrates of the streams are shaped at the encoder level to avoid peak delays for the signals. Because all the controls are centered at the application layer, we can constrain the signal latencies to be below the QoS requirements. As illustrated in Fig. 3.4, we propose an application layer transmission protocol that multiplexes the incoming signals from a teleoperation

session and provides the lowest possible latency based on the estimated available transmission rate. The scheme assigns time stamps to audio-video and haptic signals to measure the latency of each modality and detects the QoS violations to converge back to the desired delay constraints. The signals are multiplexed into a single stream to reduce the header usage at the transport layer and to control the overall system throughput under bandlimited network conditions. To measure the available transmission rate, the scheme inserts time stamps for each packet, and the demultiplexer acknowledges them back to the multiplexer side to estimate the network capacity. Once the transmission rate is estimated, the overall system throughput and video bitrate are updated accordingly. When the OP and TOP sides are geographically far away, two-way propagation or round trip time (RTT) delay is unavoidable. As the RTT delay increases, the reaction of the multiplexing scheme to network transmission rate drops is delayed. Thus, the system continues providing a high data rate to the communication pipeline as a result of delayed transmission rate estimation. The congestion event can be modeled using Eq. 3.8. When the available transmission rate TR decreases, the packets are queued at the network node where the traffic load increases. Therefore, during the congestion period, the service time of the packets goes beyond the target delay, and the QoS constraints of the teleoperation signals are violated. The proposed scheme detects the transmission rate drops and recovers the system from an overflow situation by adapting its throughput rate and video encoder bitrate.

Among the related approaches at the application layer, [ECES11, ITN11, YTY13, YYYK14] considered haptic communication scenarios over transmission links with a constant and known capacity. Their common strategy was to buffer and skip the streams to fit into the communication pipeline. In contrast to these research works, we employed advanced audio-video encoders with real-time frame-level rate-shaping capabilities [VTMM10, GCE+15], and we used an intelligent method of skipping haptic data, the perceptual deadband-based haptic data reduction scheme [HHC+08]. Moreover, we shifted the transmission rate estimation methods from the transport layer to the application layer to adapt the bitrate of the source encoders. Further details of the multiplexing scheme are discussed later in Chapter 5 of this thesis.

3.4 Chapter summary

We detailed the considered teleoperation scenario and analyzed the multimodal channel-sharing problem, which this thesis focuses on solving. The following items are the most important aspects to highlight:

- The scheduling problem concerning video packets and haptic packets was illustrated, and possible solutions were suggested.

- The benefits of using an accurate video bitrate controller were demonstrated based on both theoretical reasons and practical simulations.
- The essential blocks of the proposed multiplexing scheme were introduced. Additionally, the scheme's advantages and contributions were emphasized and compared to related research work.

Chapter 4

Rate Control for Low-Delay Video Communication

In low-delay interactive applications, such as teleoperation, remote vehicle control, online cloud gaming and videoconferencing, the latency and the quality of the visual content can be critical if the video stream is transmitted over a communication link under a rate constraint. Hence, the rate control (RC) during the video coding is a very important factor for satisfying the rate constraints of the transmission medium.

In recent decades, digital video encoding has seen accelerated progress and has reached a mature state. This period yielded many video coding standards with advanced codec structures. One of the most popular standards is H.264/AVC [ITU05], which is widely used for several audio-visual services. In contrast to preceding standards, such as MPEG4 and H.263, H.264/AVC is able to encode the frame by partitioning it into variable block sizes and perform the motion estimation with sub-pel accuracy to achieve enhanced visual quality at the motion boundaries. Furthermore, the usage of multiple reference frames improves the visual quality for inter predictions with significant rate-distortion (RD) performance. On the other hand, the computational complexity of coding one frame is increased with the new encoding features and modes. To make the codec available for practical applications, Merritt and Vanam implemented a CPU-optimized version of the standard with the name x264, described in [MV06], which can encode video frames at high-definition resolutions in real time.

Video RC schemes are not included in the released video encoding standards. However, they play a critical role in many interactive video applications. ρ-domain RC [HM02b] is one of the most efficient RC methods for transform coding-based video codecs. It relies on a linear relationship between the number of zero transform coefficients after quantization and the overall picture size. Furthermore, Zhang and Steinbach improved the ρ-domain RC scheme by extending the model with the estimation of encoding headers [ZS11]. Under this approach,

the quality of the video sequence is enhanced, and the output stream better achieves the target bitrates. However, the ρ-domain RC approach increases the computational complexity of encoding a video frame. In this chapter, we focus on reducing the encoding time with new RC features for the ρ-domain RC approach to improve the overall latency and visual quality of the interactive video stream.

This chapter proceeds as follows. We review the related work in Section 4.1. Section 4.2 introduces the proposed RC scheme [GCE+15] based on ρ-domain RC in [HM02b, ZS11]. We present the experimental results in Section 4.3, and finally, Section 4.4 summarizes this chapter.

4.1 Related work

When we consider an interactive video application over a rate-constrained link, the video frames need to be encoded and transmitted as fast as possible. Endoh formulated this problem in [EYY08] as the following minimization problem:

$$t_{enc} + \frac{s_{frame}}{TR} \rightarrow min \tag{4.1}$$

where t_{enc} describes the encoding time for a single frame. The available transmission rate is given as TR, and the size of a frame is called s_{frame}. Together, this is the time required to encode and transport a single frame. By analyzing the formula, we can deduce two facts:

1. Larger frames result in an increased transmission time.
2. t_{enc} directly affects the total processing time.

Altogether, the minimum describes our target for achieving an optimized encoding process with a minimal encoding and transmission time. Moreover, the human eye requires a minimum frame rate f to have a natural motion perception. To process the video with a sufficient number of frames per second, the following conditions need to be satisfied at all times:

1. $t_{enc} \leq \frac{1}{f}$ to ensure that the target frame rate f is achieved at all times.
2. $s_{frame} \leq \frac{TR}{f}$ to avoid buffer overflow in the communication medium.

In cases where one of the two rules is violated, subsequent frames might be dropped to guarantee real-time video transmission. This occurs not only if the encoding time is too high but also if the frame size is too large. The latter problem was introduced as buffer overflow in [RCL99]. RC algorithms attempt to avoid this problem by adjusting the quantization parameter (QP) such that the size of the compressed image remains below the upper limit. It is also important that the available bits are used. If there are too many unused bits, several

macroblocks (MBs) and frames could have been encoded with a lower QP, which results in a higher image quality. This effect is called buffer underflow. Altogether, the objective of an RC algorithm is to efficiently use the available transmission rate and to avoid the mentioned issues.

In Fig. 4.1, RC at the MB level is shown. The encoder first applies motion estimation and compensation, and then, it transforms the residual signal. There is a chicken-egg relation between the rate and the QP parameter. The encoder needs to decide which QP generates the target rate for the current MB. To solve this problem, we need a model that represents the shape of the RD function.

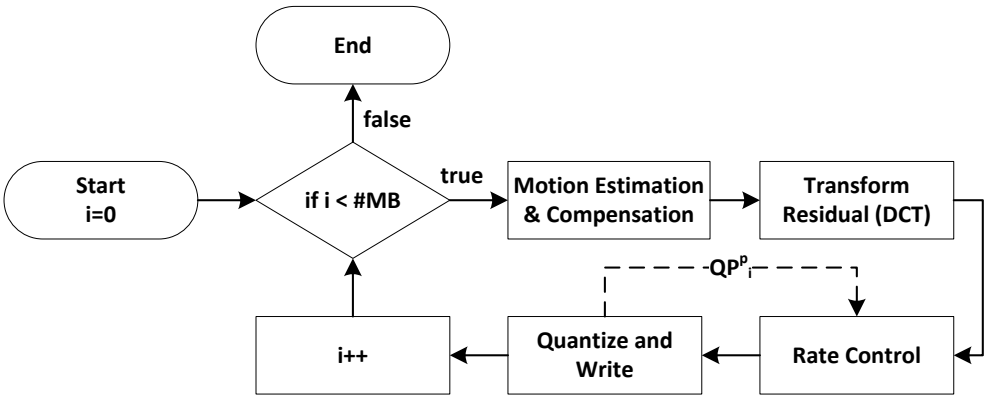


Figure 4.1: Rate control chain at the MB level: For each MB, motion estimation and compensation are applied, and the residual signals are transformed into DCT coefficients. The "Rate Control" block performs the rate distortion optimization (RDO) process to select the QP parameter and encoding mode for the target bitrate R^i_{MB} of the MB_i. However, a model for the QP selection is needed to achieve the target bitrate, as illustrated with the dashed feedback to the Rate Control block.

For example, the following Eq. 4.2 is the quadratic approximation of the RD function given for one of the RC algorithms [LCZ97, LCZ00] that was introduced as part of the MPEG-4 standard.

$$\frac{R_i - H_i}{M_i} = \alpha \cdot \frac{1}{Q_i} + \beta \cdot \frac{1}{{Q_i}^2} \tag{4.2}$$

where R_i is the target bitrate. The MB header size is also considered in this equation as H_i, and it is estimated from the previous MBs. Q_i is the QP and is used here to describe the distortion. α and β are constants that depend on the content of the video stream and are updated after every single frame. M_i is the mean absolute difference (MAD) of the current MB, calculated after performing motion estimation by only using the luminance values without residual encoding. The Q_i parameter is estimated using the model above, and the current MB is quantized with this QP to reach the target R_i.

4.1.1 ρ-domain rate control

In [HM02b], Z. He and S. K. Mitra introduced a linear model named ρ-domain RC, which relates the total bitrate required to encode a frame and the percentage (ρ) of the zero DCT coefficients after quantization in a frame. In Eq. 4.3, the linear model is given as

$$R(\rho) = \theta \cdot (1 - \rho) \tag{4.3}$$

where $(1 - \rho)$ is the percentage of non-zero DCT coefficients after quantization and θ is the model parameter related to the image content. θ is computed as the average number of bits required to encode the non-zero DCT coefficients within a frame. The fraction of zeros, ρ, increases with increasing QP. In [HM02b], the one-to-one mapping between ρ and QP is given as follows:

$$\rho(QP) = \frac{1}{S} \sum_{|x|<\Delta} P(x) \tag{4.4}$$

where $P(x)$ is the distribution of the un-quantized coefficients, S is the total number of transform coefficients in the frame, and Δ is the deadzone of the quantizer, which is determined by QP. The complexity within a frame is almost never evenly distributed across all MBs. To ensure that more complex MBs are allowed to use more bits, it is necessary to know how much information every MB contains before encoding the first MB. Therefore, a two-stage scheme should be used, as shown in Fig. 4.2.
Using Eqs. 4.3 and 4.4, the two-stage RC [HM02b] is applied with the following steps:

1. **Determine frame-level statistics:** For all MBs in the current frame, apply motion compensation, intra prediction and block transform. Determine the distribution of the transform coefficients $P(x)$.

2. **Determine QP for the current MB:** Find the target fraction of zero DCT coefficients (ρ) for the remaining MBs based on the remaining bit budget R_{left} using Eq. 4.3. Using the one-to-one relation between ρ and QP (Eq. 4.4), determine the QP for the current MB.

3. **Update parameters:** Apply encoding to the current MB with the QP obtained at item 2. Update θ in Eq. 4.3 using ρ and the number of produced bits by the current MB. Remove the used transform coefficients in the current MB from the distribution $P(x)$.

4. **Loop:** Repeat items 2 and 3 until all MBs in the frame are encoded.

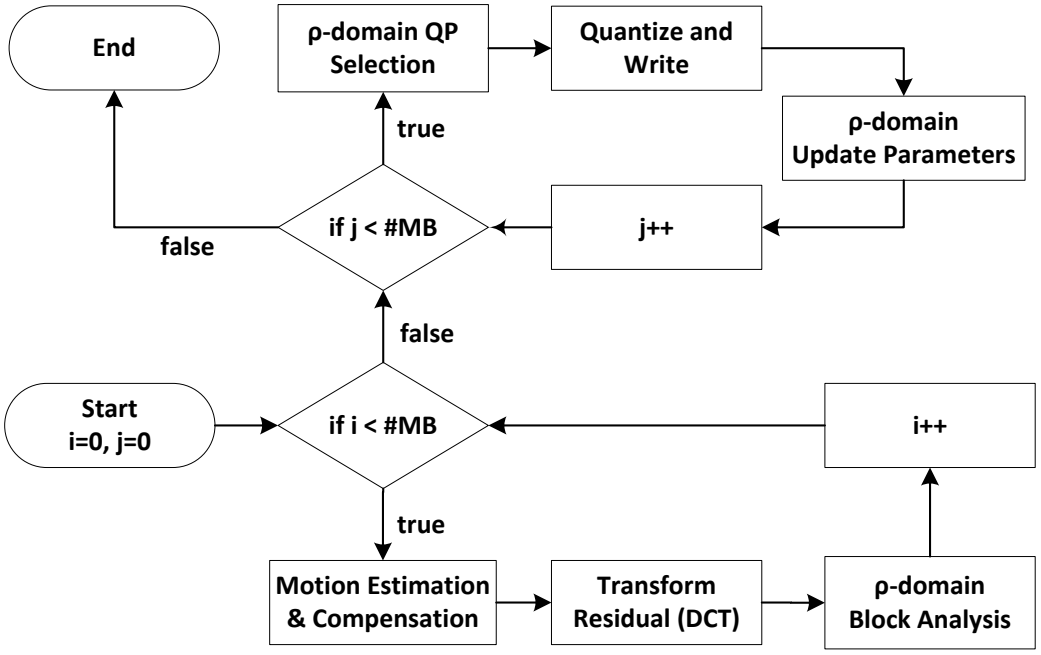


Figure 4.2: MB-based encoding scheme including ρ-domain RC: The first-stage of the encoding flow determines the ρ-QP distribution, and the second stage performs the actual encoding using the optimum QP for the target rate.

Considering the complex quantization process in H.264/AVC, it is difficult to determine the one-to-one mapping between ρ and QP. To achieve the optimum mapping, the transform coefficients for all MBs in a frame are quantized using all possible QP values. The results of stage 1 are stored in a lookup table, and a proper (ρ, QP) pair is searched for a given ρ in stage 2. This process demands many computations and increases the encoding time of a frame. In the following section, we introduce an improved version of the ρ-domain RC algorithm that can capture the relationship between ρ and QP at the MB level. Further improvements are also shown for determining QP to avoid deviations of the frame size from the target bitrate.

4.2 Proposed MB-level rate control algorithm

The linear rate model defined in Eq. 4.3 can also be applied at the MB level. Fig. 4.3 shows the percentage of non-zero coefficients, $(1-\rho)$, and the size of a MB for the CIF resolution Foreman video sequence encoded at 400 $kbps$. As observed from the figure, the linear model maintains its accuracy at the MB level. In H.264/AVC, the quantization step, Q_{step}, has the following relation with QP:

$$Q_{step} = 2^{\frac{QP-4}{6}} \tag{4.5}$$

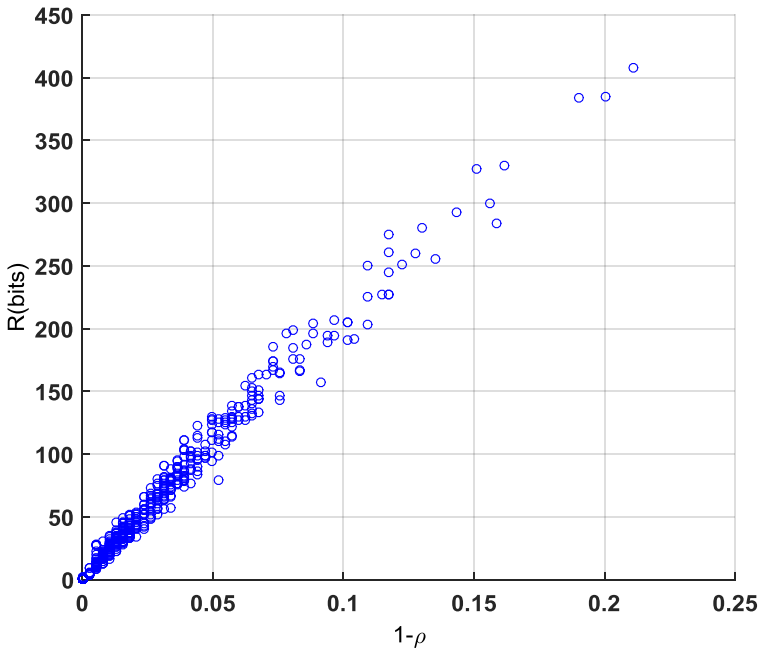


Figure 4.3: Relationship between R and $1-\rho$ at the MB level for the Foreman video sequence (CIF resolution encoded with x264 at 400 $kbps$).

When we combine this relation with the percentage of non-zero coefficients, $(1-\rho)$, the following exponential relation is obtained, as shown in Fig. 4.4. This exponential relation can be fitted by the following mathematical model in Eq. 4.6:

$$\rho = 1 - a \cdot e^{b \cdot Q_{step}} \tag{4.6}$$

Hence, the fitted model can represent the relationship between ρ and QP. Therefore, this model can be utilized to find the proper (ρ, QP) mapping instead of applying a full search over all possible QP values. Using the model in Eq. 4.6, the RC can be performed after the estimation of the unknown parameters a and b in the model. First, rate distortion optimization (RDO) is performed using the average QP of the previous frame for all MBs in the current frame. Then, the transform coefficients under the best chosen mode are quantized with two known QP values (QP_1 and QP_2). For each MB, the corresponding ρ_1 and ρ_2 are computed. The following equation set is formed to estimate the model parameters a and b.

$$\begin{aligned} \rho_1 &= 1 - a \cdot e^{b \cdot Q^1_{step}} \\ \rho_2 &= 1 - a \cdot e^{b \cdot Q^2_{step}} \end{aligned} \tag{4.7}$$

Once the model parameters are obtained, we can compute the corresponding QP for any given ρ value. Table 4.1 provides the Pearson correlation coefficient ($r_{X,Y}$) between the actual (X)

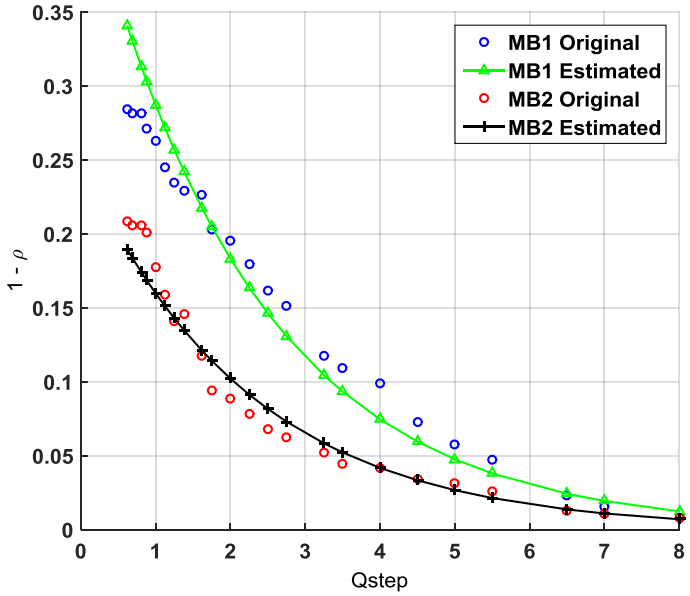


Figure 4.4: Relationship between $Qstep$ and $1-\rho$.

and estimated (Y) values for the selected test sequences, which is computed as follows:

$$r_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{4.8}$$

where μ_X, μ_Y, σ_X and σ_Y are the mean and standard deviations for the actual and estimated values. From the table, it can be observed that the correlation between the actual data and the estimated data is greater than 0.9, which indicates that the exponential model accurately estimates QP from ρ.

Table 4.1: Pearson correlation coefficients between the actual and estimated values for the percentage of texture bits $(1-\rho)$.

Sequences	Bitrate (kbps)	Correlation coefficient
Football	400	0.917
	1000	0.983
Foreman	400	0.970
	1000	0.951
Mobile	400	0.969
	1000	0.981

4.2.1 Bit allocation at the frame and MB level

Considering the low-delay video communication constraints, a constant bit budget is allocated equally to all frames per second. In the following Eq. 4.9, R_T is the bit budget per frame, BR is the available bitrate in $bits/second$, and f is the frame rate of the video sequence in $frames/second$.

$$R_T = \frac{BR}{f} \tag{4.9}$$

Next, the bit budget R_T needs to be fairly distributed among the MBs to obtain the minimum distortion. To achieve a fair budget allocation until the MBs near the end of a frame, the MB-level bit allocation scheme in [JL06] is used, as shown in Eq. 4.10.

$$R^i_{MB} = (\omega_1 \cdot \frac{R_{left}}{N_{left}} + \omega_2 \cdot avg\_R_{MB}) \cdot \frac{MAD_i}{MAD_F} \cdot S_i \tag{4.10}$$

where R^i_{MB} is the number of bits spent for MB_i at position i; R_{left} and N_{left} are the number of remaining bits and the number of un-coded MBs in a frame, respectively; avg_R_{MB} is the average target number of bits for each MB, which is given by Eq. 4.11;

$$avg\_R_{MB} = \frac{R_T}{N_{MB}} \tag{4.11}$$

where N_{MB} is the total number of MBs within a frame; MAD_i is the mean absolute difference (MAD) of MB_i; MAD_F is the MAD of the current frame; and S_i is the position-dependent scaling factor, which is given by Eq. 4.12.

$$S_i = \alpha_0 \cdot \frac{i}{N_{MB}} + \alpha_1 \tag{4.12}$$

where α_0 and α_1 are constants, which are set as 0.4 and 0.8, respectively. As suggested in [JL06], the weights ω_1 and ω_2 in Eq. 4.10 are set as 0.7 and 0.3, respectively. Given the allocated bits R^i_{MB} for MB_i, the number of texture bits $tex\_R^i_{MB}$ for this MB is given by the following Eq. 4.13.

$$tex\_R^i_{MB} = R^i_{MB} - R^i_{hdr} \tag{4.13}$$

where R^i_{hdr} is the estimated number of header bits for MB_i, which is obtained by averaging the number of header bits generated by all previously coded MBs in the current frame.

4.2.2 QP determination at the MB level

When the pure texture bits, $tex\_R^i_{MB}$, for MB_i are computed, the percentage of zero coefficients ρ_i among the quantized transform coefficients in MB_i can be determined using Eq. 4.3. Then, using Eq. 4.6, the quantization step $Qstep_i$ for MB_i is computed. Finally, the corresponding QP_i is obtained with Eq. 4.5. To ensure that the spatial quality remains smooth

Algorithm 1: Switched calculation of QP.

```
1 if R_left ⩾ thr then
2     QP_i is calculated using Eqs. 4.5, 4.6 and 4.14
3 else
4     QP_i is set as QP_{i-1} + 4
```

within a frame, QP_i should be bounded. In our framework, we adopted the QP adjustment scheme proposed in [JL06] as follows:

$$QP_i = min\{QP_{i-1} + \Delta QP, max\{QP_i, QP_{i-1} - \Delta QP\}\} \tag{4.14}$$

where QP_{i-1} is the QP of MB_{i-1} and ΔQP is the varying range of QP along the MBs. The initial value of ΔQP is 2, and after encoding each MB, it is updated for the QP determination of the next MB as follows, i.e., MB_{i+1}:

$$\Delta QP = \begin{cases} 1, & if\ QP_i \geq 25 \\ 2, & otherwise \end{cases} \tag{4.15}$$

In low-delay video communication, the buffer size is very small. Therefore, we apply a threshold to control the QP calculation. With this, we avoid large deviations of the frame size from the target bit budget. The thresholding is defined as follows:

$$thr = n \cdot \frac{prev\_R_{hdr}}{prev\_R_{total}} \cdot R_{left} \tag{4.16}$$

where n is a constant, $prev\_R_{hdr}$ and $prev\_R_{total}$ are the header bits and the total number of bits produced by the previous frame, respectively; and R_{left} is the remaining bits for the uncoded MBs in the current frame. Hence, a switched calculation of QP is applied, as illustrated in Algorithm 1.

4.2.3 Summary of the rate control algorithm

In Algorithm 2, we give the sequential order of the computations in the proposed RC scheme. As already shown in Fig. 4.2, the encoding is conducted in two stages.

1. The first stage is given between lines 3 and 17 in Algorithm 2, where we apply analysis on the MBs and RDO. For each MB, the motion estimation, compensation and mode decision are made (lines 12-17). During the mode decision process, if the frame is encoded as a P-frame, the average QP of the previously encoded frame is used (line 11). For the I-frames, a constant QP is applied (lines 4-8). At the first stage, motion information and MADs for the best mode are stored for the QP estimation at the second stage.

Algorithm 2: MB-level RC algorithm.

1 //* **Frame-level bit budget computation (Eq. 4.9)** *//
2 $R_T = Bandwidth/framerate$
3 /**** **STAGE 1: Analysis** ****/
4 **if** $CurrentFrame == Intra$ **then**
5 **if** $bitsperpixel \geq 0.13$ **then**
6 $QP_{init} = 30$
7 **else**
8 $QP_{init} = 45$
9 **else**
10 //* **For inter-frames, assign average QP of the previous frame** *//
11 $QP_{init} = AvgQP_{f-1}$
12 **while** $MB_i != EndOfFrame$ **do**
13 //* **Perform RDO on every MB** *//
14 //* **MVs and MAD for the best mode for each MB are recorded** *//
15 $MV_i = MotionEstimation(CurrentFrame, PreviousFrame)$
16 $MotionCompensation(PreviousFrame, MV_i)$
17 $(MAD_i, Mode_i) = ModeDecision(MB_i, QP_{init})$
18 /**** **STAGE 2: Rate Control** ****/
19 $avg\_R_{MB} = R_T/N_{MB}$ //* **average rate per MB Eq. 4.11** *//
20 **while** $MB_i != EndOfFrame$ **do**
21 //* **Using Eqs. 4.10, 4.12 and 4.13, find pure texture rate for** MB_i *//
22 $S_i = \alpha_0 \cdot i/N_{MB} + \alpha_1$
23 $R^i_{MB} = (\omega_1 \cdot (R_{left}/N_{left}) + \omega_2 \cdot avg\_R_{MB}) \cdot (MAD_i/MAD_F) \cdot S_i$
24 $tex\_R^i_{MB} = R^i_{MB} - R^i_{hdr}$
25 //* **Using Eqs. 4.3 and 4.7, determine rate model parameters** *//
26 $(a, b) = ComputeRateModelParameters(\rho_1, \rho_2, Q^1_{step}, Q^2_{step})$
27 //* **Find the final** QP_i **using Eqs. 4.14, 4.15, and 4.16 and algo. 1** *//
28 $QP_i = ComputeQuantizationParameter(tex\_R^i_{MB}, \theta, a, b)$
29 //* **Perform actual encoding for** MB_i *//
30 $R^i_m = EncodeMacroblock(MB_i, QP_i)$//* R^i_m **is the resulting rate for** MB_i *//
31 $R_m = R_m + R^i_m$ //* **Update the used bits so far** *//
32 //* **Update the model parameter** θ *//
33 $\theta = R_m/(384 \cdot N_m - N_{zero})$

2. In the second stage, the final QP is determined for each MB using the proposed MB-level ρ-domain model, and then, the actual encoding is applied. Although we break up the original codec structure of the standard into two stages, the motion estimation and mode decision are performed only once. This allows the proposed MB-level RC scheme to have a similar computational complexity as one-pass RC algorithms. In Algorithm 2, the second stage is given between lines 18 and 33. First, the average rate per MB is computed in line 19; then, the encoding is processed for each MB. Between lines 21 and 24, the pure texture bitrate, $tex\_R^i_{MB}$, for MB_i is determined. At line 26, MB_i

is encoded using two candidate Q_{step} values to obtain the model parameters a and b in Eq. 4.6. The corresponding QP_i for MB_i is computed in line 28 using the rate model parameters, (a,b,θ), and the target texture bits, $tex\_R^i_{MB}$. The RC algorithm performs the actual encoding at line 30 and writes MB_i to the bit stream. R^i_m is the total number of bits required to encode MB_i. Finally, at line 33, after encoding MB_i, the model parameter θ should be updated using the following equation:

$$\theta = \frac{R_m}{384 \cdot N_m - N_{zero}}$$

where N_m is the number of coded MBs in the current frame, R_m is the number of bits produced by these coded MBs, and N_{zero} is the number of zero coefficients in these coded MBs. Note that there are 384 coefficients for a MB in YUV 4:2:0 format.

4.3 Experimental results

The proposed RC scheme is implemented in x264 [MV06]. The encoder is configured to conform to the baseline profile. Context-adaptive variable-length coding (CAVLC) is used for entropy coding, and there is only one reference frame for each inter-coded frame to avoid high computational complexity and increased encoding time. For comparison purposes with the video compression literature, we select the commonly used CIF (352×288) format sequences Bus, $Container$, $Football$, $Foreman$ and $Mobile$ as a test set, whose frame rates are all 25 *fps*. These five test sequences are selected because they represent different levels of spatial and temporal complexity. For each sequence, 250 frames are encoded, in which the first frame is compressed as the I frame type (intra only) and the remaining frames are encoded as P frames.

We compare the proposed RC algorithm [GCE+15] with the original ρ-domain RC algorithm in [HW08], in which the transform coefficients are quantized by all possible QP values to obtain the (ρ, QP) table; then, all possible QP values are checked to select the proper QP for a given ρ. To perform a fair comparison, the proposed RC scheme and the original ρ-domain RC both adopt the frame-level bit allocation and initialization of QP for RDO presented in Section 4.2.3.

Moreover, we performed real-time streaming experiments to illustrate the bitrate accuracy and delay-jitter performance of the proposed RC algorithm in comparison with the original constant bitrate RC scheme (CBR) in x264.

4.3.1 Video quality in terms of PSNR

Table 4.2 lists the PSNR and bitrate (BR) of the proposed RC ("*Proposed*") and the original ρ-domain RC ("*Original*"). From Table 4.2, it can be observed that the proposed method can

achieve better visual quality (PSNR) than the original method for most of the test sequences because the proposed method adopts the MB-level bit allocation, which can improve the frame quality by properly distributing the bits among all MBs. It is observed that, for the *Football* sequence, the proposed method performs at low bitrates, worse than the original method in terms of PSNR because the spatial and temporal contents of the *Football* sequence are very complex. Additionally, we can conclude that the effect of MB-level bit allocation is reduced at low bitrates due to the limited target bit budget.

4.3.2 Bitrate accuracy of rate control

From Table 4.2, we observe that the actual number of bits produced by the proposed method is much closer to the target bitrate when compared to the original method. This improvement is due to the switched QP calculation scheme in the proposed method. Table 4.2 presents the average deviation of the frame size from the target bit budget, R_T, for each sequence, which is calculated as follows:

$$Dev = \frac{1}{T} \cdot \sum_{j} \frac{|R^j_{actual} - R_T|}{R_T} \cdot 100\% \tag{4.17}$$

where R^j_{actual} is the frame size produced by frame j, R_T is the target bit budget of each frame, and T is the total number of encoded frames in a sequence. From Table 4.2, it can be observed that the proposed method has the smallest deviation, which indicates that it can control the frame size more accurately and fully utilize the transmission capacity of the channel.

4.3.3 Computational complexity

In the original ρ-domain RC, the transform coefficients are quantized by all possible QP values to obtain the (ρ, QP) table. For a given ρ, the RC checks all possible QP values to select the proper QP. In the proposed method, the transform coefficients are only quantized with two QPs to calculate the model parameters in Eq. 4.6. For a given ρ, the QP can be calculated with the method in Section 4.2.2. Hence, the computational complexity of the proposed method is expected to be lower than that of the original approach. Here, we use the reduction in the encoding time to determine the computational complexity of the two methods, which is defined as follows:

$$\Delta_C = \frac{C_{Org} - C_{Pro}}{C_{Org}} \cdot 100\% \tag{4.18}$$

where C_{Org} and C_{Pro} are the encoding times of the original method and the proposed method, respectively. The reduction in the encoding time for each sequence is shown in Table 4.2. From Table 4.2, it can be observed that the reduction in the encoding time is between 40% to 58%. Therefore, the proposed method is better suited for low-delay video communication.

Table 4.2: Performance comparison between the proposed algorithm [GCE+15] and the original algorithm [HW08] in terms of average bitrate, standard deviation of the bitrate, PSNR and encoding time reduction (Δ_t).

Sequences	Target BR (kbps)	R_T (bytes)	Original			Proposed				
			BR (kbps)	Dev[%]	PSNR (dB)	BR (kbps)	Dev[%]	PSNR (dB)	PSNR Gain (dB)	Δ_t (%)
Bus	300	1500	242.86	28.05	25.29	293.40	**2.94**	**25.75**	0.46	52.79
	500	2500	501.81	7.18	27.28	490.22	**2.55**	**27.45**	0.17	48.64
	1000	5000	995.57	2.12	28.90	985.75	**1.45**	**29.60**	0.70	46.06
	2000	10000	1998.32	0.81	30.12	1981.65	**0.65**	**33.41**	3.29	45.13
Container	300	1500	237.09	21.61	34.11	292.32	**3.63**	**34.30**	0.19	45.90
	500	2500	441.88	11.93	35.68	494.86	**1.72**	**36.50**	0.82	45.72
	1000	5000	946.73	5.47	37.21	990.90	**1.08**	**39.01**	1.80	42.79
	2000	10000	1962.52	1.95	38.10	1997.27	**0.60**	**40.78**	2.68	44.39
Football	300	1500	228.21	24.62	**26.51**	295.42	**2.30**	26.35	-0.16	58.17
	500	2500	450.80	10.26	**29.11**	493.61	**1.82**	29.03	-0.08	52.87
	1000	5000	962.33	3.92	32.38	989.38	**1.21**	**32.78**	0.40	50.29
	2000	10000	1968.35	1.67	35.80	1977.37	**1.06**	**37.12**	1.32	45.44
Foreman	300	1500	216.62	28.37	30.82	306.21	**1.94**	**32.00**	1.18	49.82
	500	2500	443.36	11.77	33.74	508.10	**1.53**	**34.21**	0.47	49.03
	1000	5000	952.23	4.94	35.76	1004.56	**0.76**	**36.70**	0.94	42.50
	2000	10000	1957.95	2.17	37.13	2000.21	**0.42**	**38.99**	1.86	40.28
Mobile	300	1500	251.08	17.56	24.07	298.95	**1.61**	**24.47**	0.40	56.67
	500	2500	460.68	8.60	25.63	497.75	**1.41**	**26.66**	1.03	56.00
	1000	5000	968.48	3.39	27.43	995.47	**1.17**	**29.81**	2.38	52.22
	2000	10000	1974.51	1.37	29.13	1990.37	**0.49**	**33.37**	4.24	48.40

4.3.4 Real-time transmission tests

In this experiment, we show the real-time performance of the original RC mode of x264, which is called x264-CBR [MV07], and the proposed MB-level ρ-domain RC scheme (x264-RHO) for a teleoperation video recorded at 720p (1280×720) resolution at 25 fps. This experiment investigates the delay-jitter performance of the RC algorithms when the video encoder and decoder are separated by a communication link having a constant bitrate capacity. In Sections 3.2 and 3.3.1, detailed delay models have been discussed, and the benefit of using bitrate control schemes for video compression has been shown. This experiment demonstrates how the waiting time in queue W_q described in Eq. 3.4 is minimized by controlling the video frame sizes accurately.

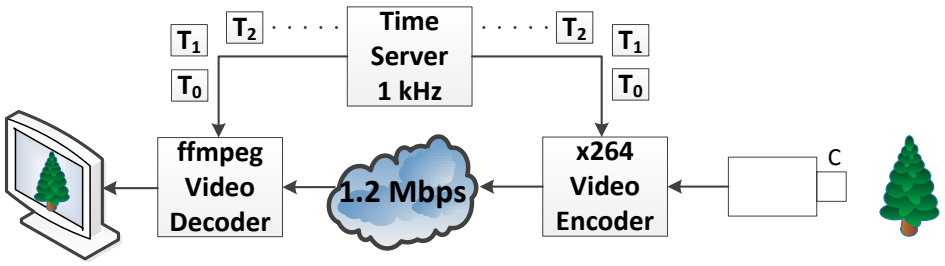


Figure 4.5: Real-time video streaming testbed.

Figure 4.6: Teleoperation video 720p @ 25 fps recorded by an eye-on-the-hand GigE machine vision camera.

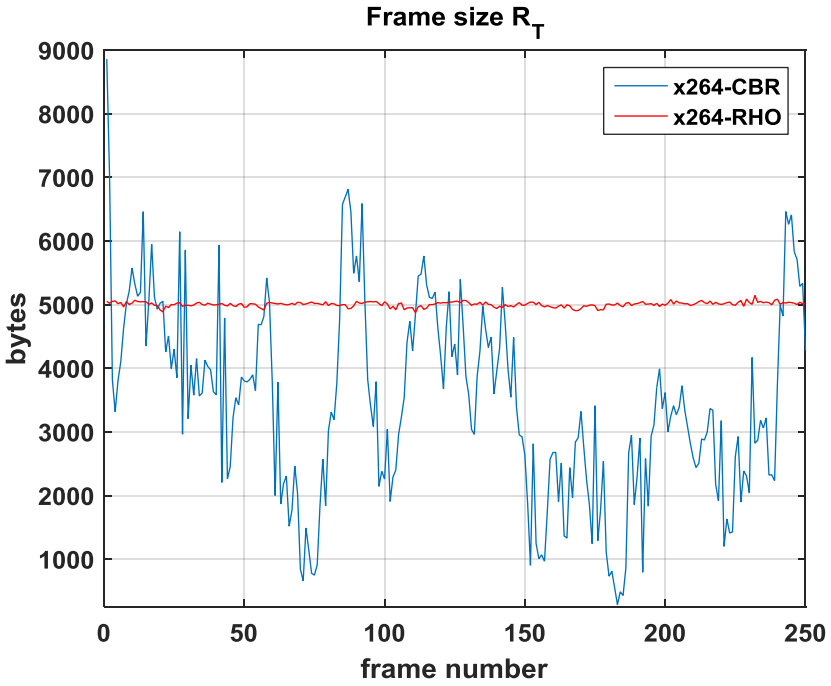
To perform the experiment, a video communication system is built, as illustrated in Fig. 4.5. For both RC schemes, the target video bitrate is set to 1 $Mbps$, and the compressed stream is transmitted over a 1.2 $Mbps$ CBR link. There is a time server connected to both sides, and it sends time stamps with a 1 ms period to achieve synchronized time-delay measurement of the video frames. When the frame is captured or read from the file, it is immediately passed to the x264 video encoder block. Whenever a frame is encoded, it is transmitted over the

communication link together with the recent time stamp received from the time server. As the frame arrives at the video decoder side, the transmission delay is measured by subtracting the recent time stamp received at the decoder from the time stamp of the current frame.

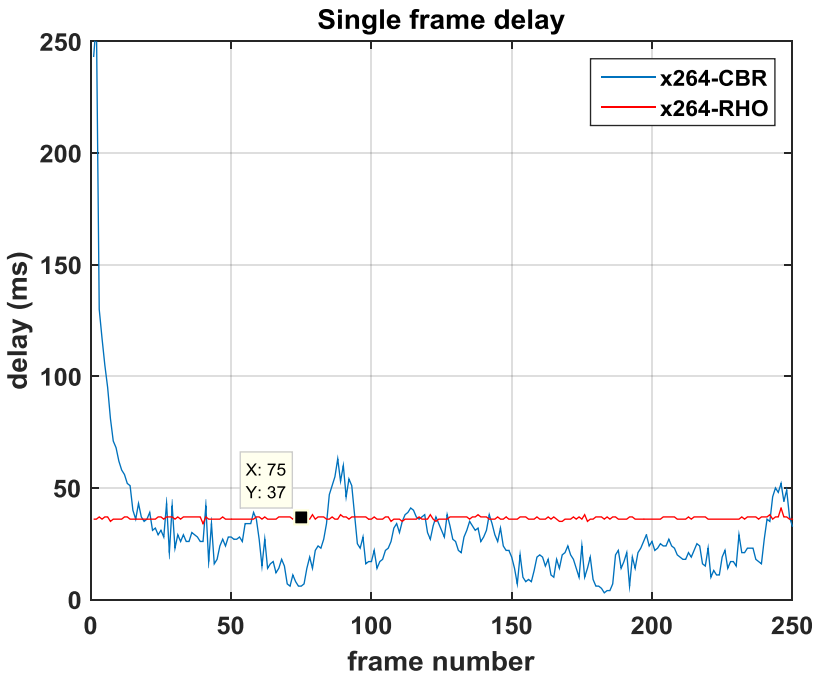
Fig. 4.7(a) presents the comparison between the proposed MB-level ρ-domain RC (x264-RHO) and the constant bitrate mode of x264 (x264-CBR). It can be observed that the frame size fluctuation of x264-CBR is much larger than that of x264-RHO, which proves the advantage of the proposed RC scheme. For a 1 $Mbps$ video with a frame rate of 25 fps, each frame should be 5000 $bytes$. As observed from Fig. 4.7(a), x264-RHO can accurately control the frame size to the target frame size. In Section 3.2, we described the queuing delays due to the variable video frame size and showed that the variation in service time (jitter) and waiting time in queue can be reduced with frame-level accurate bitrate control. Fig. 4.8(b) demonstrates the queuing simplification from the D/G/1 (probabilistic) model to the D/D/1 (deterministic) model. The proposed RC scheme converges to a controllable constant delay, which is the service time of the communication link. For further statistical comparisons, Figs. 4.8(a) and 4.8(b) show the delay distributions for both RC schemes. We can conclude that the proposed RC scheme provides a low delay and almost 0 ms jitter for real-time video streaming, which makes the RC scheme a reliable technique for a teleoperation system.

4.4 Chapter summary

In this chapter, we described a MB-level RC algorithm for low-delay video communication based on the ρ-domain rate model [HM02b]. In the proposed RC scheme, an exponential model is employed to fit the relationship between ρ and $Qstep$. Hence, the QP for each MB can be obtained in an efficient way. As a further extension, a switched QP calculation scheme is developed to avoid large deviations of the actual frame size from the target bit budget. Compared with the original ρ-domain RC [HW08] for H.264, the proposed method can achieve better video quality and higher bitrate accuracy. The most important contribution of this chapter is the computational complexity reduction in the (ρ,QP) relation. Instead of applying brute force search over QP parameters, the proposed scheme performs QP determination by testing only two QP candidates. Therefore, the proposed RC scheme compresses the frames with an almost 50% lower encoding delay. Moreover, this allows us to encode 720p video streams at 25 fps in real time. In the following chapter, the proposed RC approach is employed as the base video encoder for the multi-modal multiplexing scheme, and it is extended with channel adaptability features.

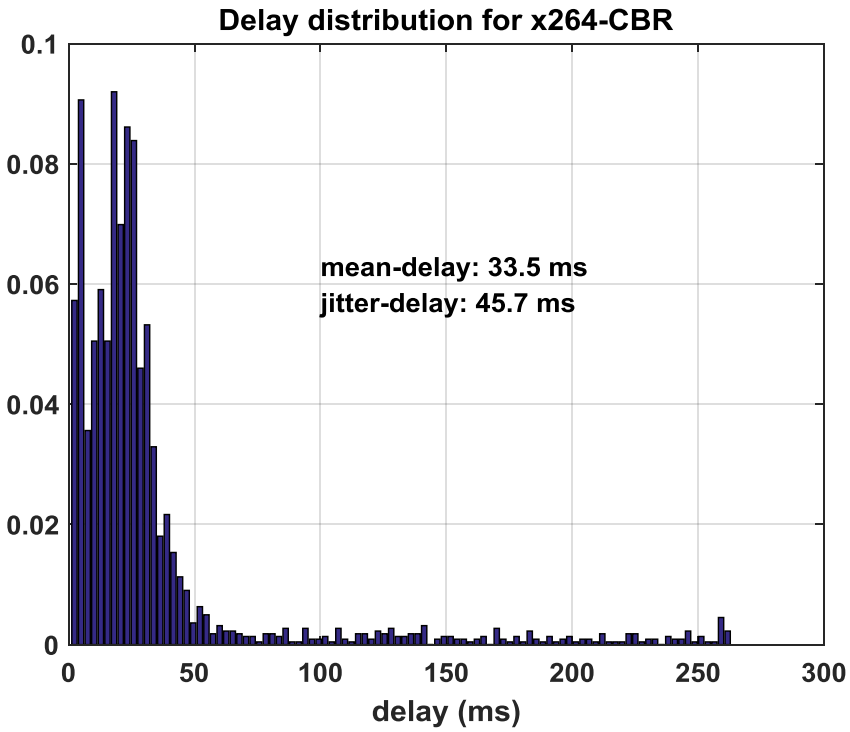


(a) Frame size comparison of x264-CBR (original RC algorithm in x264) and ρ-domain RC

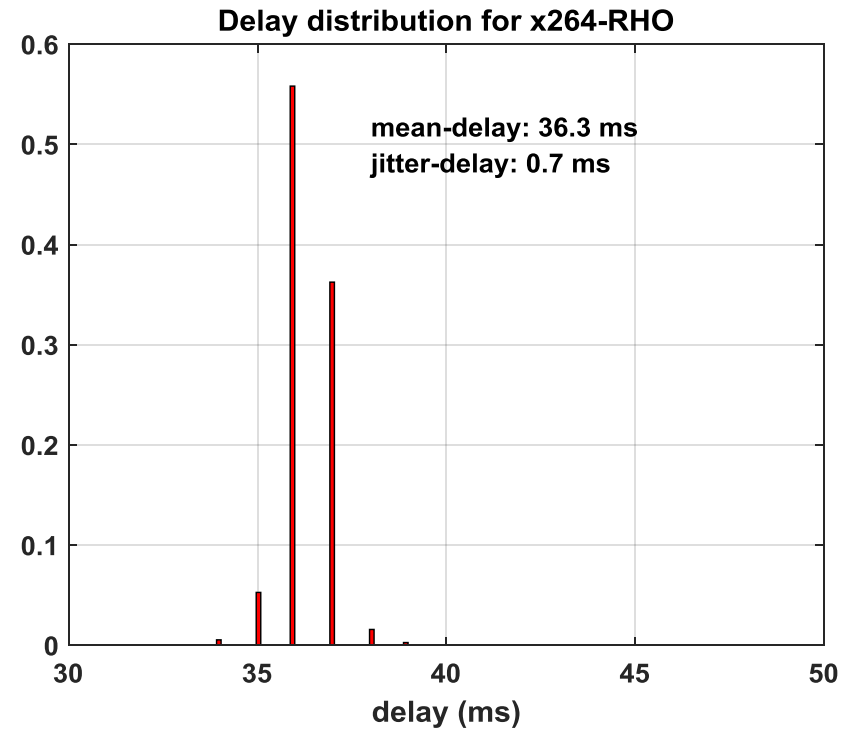


(b) Single-frame delay comparison of x264-CBR and ρ-domain RC

Figure 4.7: Experimental results for 720p @25 fps teleoperation video.



(a) Delay distribution for x264-CBR



(b) Delay distribution for x264-RHO

Figure 4.8: Delay and jitter performance of the RC schemes.

Chapter 5

Multiplexing Scheme for Multimodal Teleoperation

In Chapter 3, we defined the low transmission rate problem and the challenge of resource allocation between audio, video and haptic signals. This chapter addresses the aforementioned issues of transmitting audio, video and haptic signals for teleoperation scenarios over communication links having low transmission rate and proposes an application layer protocol to multiplex multimodal streams into a low-capacity link.

Fig. 5.1 illustrates the media flow from the TOP to the OP. The audio, video and force signals are captured, encoded and put into the media data queues based on the FIFO (First In First Out) principle. The multiplexer (MUX) can directly access the queues to forward multimedia streams to the channel. Whenever signals are captured at the TOP side, they are passed to the corresponding encoding block. The video frames are encoded using the MB-level ρ-domain RC scheme described in Chapter 4. The audio signal is compressed using the low-delay audio codec CELT [VTMM10], and force samples are acquired at 1 kHz and then passed to the "Haptic Data Reduction" block. A perceptually motivated haptic data reduction scheme is used together with a time-domain passivity control architecture (TDPA) [XCSS15] to reduce the high sampling rate of the force signal for transmission. As observed in Fig. 5.1, the force samples are queued in a buffer of size T, and their transmission states are monitored by the MUX for packet type and scheduling decisions. The capacity of the forward channel (from the TOP to the OP) is predicted by a transmission rate estimation algorithm to immediately adapt the multiplexing throughput rate and the video bitrate to the current transmission rate of the communication link. The transmission rate estimation relies on the acknowledgment packets sent from the demultiplexer (DEMUX) side. The MUX inserts a time stamp into every packet, and when a packet arrives at the DEMUX side, an

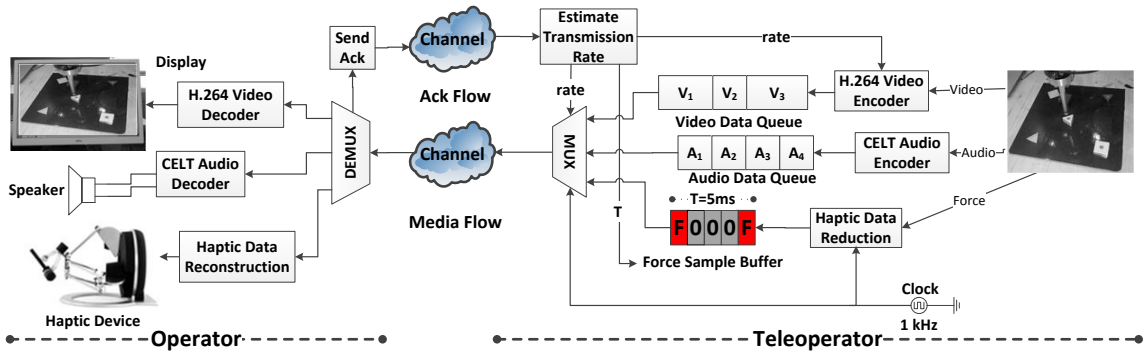


Figure 5.1: Multimodal multiplexing on the feedback channel from the TOP to the OP.

acknowledgment packet including the packet identification number and time stamp is sent back to the MUX side. The "Estimate Transmission Rate" block in Fig. 5.1 measures the time difference between the current time and the time stamp in the acknowledged packet. Using the packet size and the measured transmission delay, the available transmission rate is estimated.

When signals arrive at the DEMUX side, they are passed to the corresponding decoding block, and their time stamps, which are recorded during the generation, are fed back inside the acknowledgment packets to the MUX side. Therefore, the instant delay of each modality can be monitored as QoS parameters, which can be used to ensure that the desired delay constraints are satisfied.

The outline of this chapter is as follows:

- Section 5.1 describes the application layer protocol structure and functionality of the multiplexing algorithm with example cases.

- Section 5.2 gives the processing of a received packet and the acknowledgment preparation for transmission rate estimation and signal delay measurements. Moreover, the reaction of DEMUX is illustrated in the case of missing data streams due to packet losses in the communication link.

- Section 5.3 discusses the challenges and limitations of the transmission rate estimation problem and presents the adopted transmission rate estimation [CFM04] algorithm for predicting the true communication capacity in real time. Moreover, the negative effect of the increasing RTT due to signal propagation on transmission rate estimation is introduced, and a congestion detection and control scheme is proposed to handle sudden transmission rate drops.

- Section 5.4 defines the experimental conditions and the multiplexing scheme together with the teleoperation system, which are tested over a wide range of network conditions. The performance of the system is evaluated based on objective measures such as packet rate, end-to-end signal delay, and peak-signal-to-noise ratio (PSNR) for visual quality.

5.1 Multiplexing scheme

As introduced in Section 3.2, an appropriate scheduling strategy for audio, video and haptic signals is the preemptive and resume procedure. The preemption and resumption times depend on the haptic arrivals, which are generated by the PD-based force data reduction integrated with TDPA control scheme [XCSS15]. The constant Weber factor, k, determines the transmit/non-transmit states of the force signal. If the current force signal value exceeds the perceptual threshold defined by the previously sent sample, the encoder marks the current force sample for transmission and enqueues it into the force buffer. As previously mentioned in Section 3.3 of Chapter 3, the system needs to buffer a few force samples (see the force sample buffer in Fig. 5.1) for the observation of force transmit states. Hence, the scheme can combine streams into one packet to avoid using more protocol headers to ensure the efficient usage of network resources. As illustrated in Fig. 5.1, the "Haptic Data Reduction" and "MUX" blocks are triggered in synchrony with a clock rate of 1 kHz. When the clock ticks, the "Haptic Data Reduction" block pushes a sample to the force buffer tail, and the "MUX" dequeues a force sample from the force buffer head. Considering the transmission flag of the force sample, it is either discarded or sent. The samples tagged as red blocks "F" need to be transmitted, and the samples tagged as light gray blocks "0" do not need to be transmitted.

5.1.1 Application layer protocol structure

According to the state of the system, the MUX can generate 7 types of packets: force (F), video (V), audio (A), audio-video (AV), audio-force (AF), video-force (VF) and audio-video-force (AVF) packets. The packet types are identified by the header information, "MUX Header", which has the structure shown in Fig. 5.2. The first 3 bits marked with "M" in Fig. 5.2 represent the packet type. With 3 bits, 8 different packet types can be signaled, which is sufficient for a multiplexing scheme with three modalities. Currently, bits 3, 4, 5 and 6 ("N") are reserved for future modalities and control messages. If the packet type includes video data, bit 7, tagged as "L", is used to signal the video frame completion, which means that the frame is ready for decoding. The multiplexing scheme divides the encoded video stream into fragments of different sizes, which are determined by the irregular haptic triggers for their preemption. When the current video frame transmission is completed, the DEMUX is triggered to pass the byte stream to the video decoder.

Bit	0	1	2	3	4	5	6	7
Header	M			N	N	N	N	L

Figure 5.2: MUX header (1 byte) structure in bits.

In Table 5.1, we give the application layer protocol structure for each packet type. The MUX header is denoted by $H(1)$, where (1) denotes that the information size is 1 *byte*. According to the packet type, we need to signal additional information such as time stamps, data indexes and payload lengths after the MUX header. Every packet contains a time stamp, denoted as $PTS(2)$, which is the clock time when the packet is passed to the transport layer for transmission. Using the time stamps, the transmission time of each packet can be measured at the application layer. The multimedia information follows these headers, and according to the modality type, the following information is added:

- **Force:** If the packet contains a force sample, sample id $SID(2)$, sample time stamp

Table 5.1: Application layer packet structures.

Packet types	**Packet structure**	**Size (bytes)**
F	$H(1) - PTS(2) - SID(2) - STS(2) - SPL(6) - EPL(12)$	25
A	$H(1) - PTS(2) - NAF(1) - a \cdot (ATS(2) - AFN(2) - APLL(2)) - APL(X)$	$4 + a \cdot 6 + X$
V	$H(1) - PTS(2) - FN(2) - FGN(2) - FTS(2) - VPLL(2) - VPL(Y)$	$11 + Y$
AV	$H(1) - PTS(2) - NAF(1) - a \cdot (ATS(2) - AFN(2) - APLL(2)) - APL(X) - FN(2) - FGN(2) - FTS(2) - VPLL(2) - VPL(Y)$	$12+a\cdot6+X+Y$
AF	$H(1) - PTS(2) - NAF(1) - a \cdot (ATS(2) - AFN(2) - APLL(2)) - APL(X) - STS(2) - SID(2) - SPL(6) - EPL(12)$	$26 + a \cdot 6 + X$
VF	$H(1) - PTS(2) - SID(2) - STS(2) - SPL(6) - EPL(12) - FN(2) - FGN(2) - FTS(2) - VPLL(2) - VPL(Y)$	$33 + Y$
AVF	$H(1) - PTS(2) - NAF(1) - a \cdot (ATS(2) - AFN(2) - APLL(2)) - APL(X) - FN(2) - FGN(2) - FTS(2) - VPLL(2) - VPL(Y) - SID(2) - STS(2) - SPL(6) - EPL(12)$	$34+a\cdot6+X+Y$

$STS(2)$, sample payload $SPL(6)$ and energy payload $EPL(12)$ for the TDPA control architecture are added to the packet. Each force sample is represented by a 2-*byte* floating point number in the sample payload, and each energy sample is represented by a 4-*byte* floating point number in the energy payload.

- **Video:** If the packet contains a video frame fragment, the corresponding frame number $FN(2)$, fragment number $FGN(2)$, frame time stamp $FTS(2)$ and video payload length $VPLL(2)$ are written into the packet. The payload size (Y *bytes*) of the fragment is written into $VPLL(2)$. After the multiplexing information, the payload data are written, and they occupy Y *bytes*, where Y is determined by the multiplexing algorithm.

- **Audio:** Unlike video frames, many audio frames can fit into one packet due to its small frame size. First, we indicate the number of audio frames $NAF(1) = a$, and then, we write the time stamp $ATS(2)$, frame number $AFN(2)$ and payload length $APLL(2)$ for each audio frame. After the side information, the audio payloads (a total of X *bytes*, determined by the MUX) are written into the packet.

In addition to the multiplexing information, the data link, network and transport layers add their own header information to every packet. The used protocol(s) should be known to the application layer MUX because the protocol header size needs to be considered when determining the system throughput rate to ensure low-delay scheduling of the packets.

The teleoperation system in this thesis uses UDP/IPv4 (Universal datagram protocol and Internet protocol version 4) over ethernet. Fig. 5.3 gives the structure of an ethernet frame header. It starts with 7 bytes of preamble and 1 byte of start frame delimiter. Then, the frame includes destination and source media access control (MAC) addresses, which are each 6 bytes. Finally, a 2-*byte* ethernet type marker follows the MAC addresses. In total, each ethernet frame has 22 bytes of overhead to reveal the packet source and destination at the data link layer. Fig. 5.4 shows the header structures of IPv4 and UDP for the network and transport layers, respectively. IPv4 and UDP headers occupy 12 and 8 bytes, respectively. The application layer headers described in Table 5.1 and payload data are written into the data section of the packet.

The multiplexing scheme considers necessary overhead for the transport, network and data link layers, and it reserves 42 *bytes* for the UDP/IPv4 over ethernet-based communication.

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Fr.	Preamble							SFD	Dest. MAC Addr.						Src. MAC Addr.						Eth. Type	

Figure 5.3: Ethernet header (22 bytes).

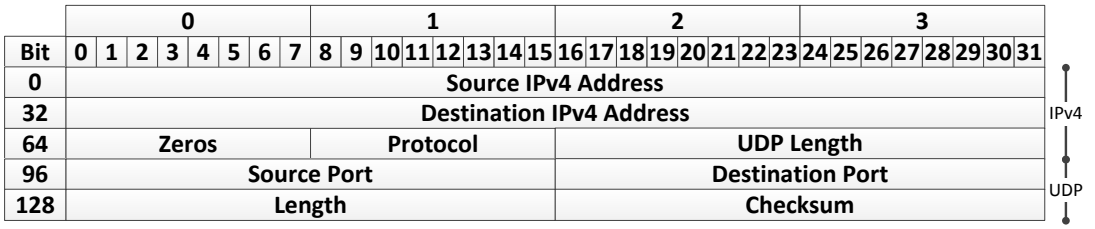


Figure 5.4: IPv4 and UDP header (20 bytes).

The overall overhead size is calculated by adding up the 42 *bytes* of the UDP/IPv4 protocol header to the estimated size of each packet type given in Table 5.1. Additionally, during the multiplexing, the maximum size of each packet is limited to the maximum transmission unit (MTU) size of the ethernet protocol: 1500 *bytes*. If the packet size at the application layer exceeds the MTU size, the ethernet protocol divides the packet into fragments. The fragmentation adds an additional overhead to the transmission and increases the packet rate. If one of the fragments is lost during the transmission, it is not possible to recover the original packet using the remaining fragments, leading to the loss of more data.

The proposed protocol manages the teleoperation session only at the application layer without interfering and modifying processes in the lower communication layers. This provides the flexibility to port the multiplexing scheme to any type of communication medium. In this thesis, we consider a teleoperation system communicating over the internet. However, it is possible to integrate the system easily into any type of communication system by simply modifying the overhead and MTU size.

5.1.2 Multiplexing algorithm

The multiplexing algorithm makes the preemption and resume scheduling decisions for packet transmission based on the force sample arrivals monitored in the fixed-length force buffer. In Algorithm 3, we give the pseudocode of the multiplexing algorithm. The MUX block runs as a thread with a clock rate of 1 kHz in synchrony with the "Haptic Data Reduction" block. The first condition statement in line 2 determines whether the channel is busy with a transmission. The $Slots = 0$ case shows that the channel is ready to serve the transmission of new data, and $Slots > 0$ indicates that the channel remains busy with the transmission of the previous packet. When the channel is in a busy period, the MUX cannot push a new packet into the channel and must wait until the channel is ready for the new transmission. When the channel comes back to the ready state, the MUX goes over the queued samples in the force buffer and counts the free slots tagged as "0" until either hitting a planned-force transmission labeled as "F" or reaching the tail of the force buffer, which occurs when there are no force transmissions.

Algorithm 3: Multiplexing algorithm.

```
1  //* Prepare next transmission when channel is free *//
2  if Slots == 0 then
3      //* Check force buffer for force transmissions *//
4      Slots = CheckAvailableSlots(ForceBuffer);
5      if Force.Transmit == "F" then
6          //* Decide packet type based on available audio and video bytes *//
7          PacketType = CheckMultimediaAvailability(AudioBytes, VideoBytes);
8          //* Merge force samples with audio and video if available *//
9          MultiplexStreams(PacketType);
10         //* Possible packet types: F, AF, VF or AVF *//
11         SendPacket(PacketType, &Slots);
12     if Force.Transmit == "0" then
13         //* Decide packet type based on available audio and video bytes *//
14         PacketType = CheckMultimediaAvailability(AudioBytes, VideoBytes);
15         //* Merge audio and video streams based on availability *//
16         MultiplexStreams(PacketType);
17         //* Possible packet types: A, V or AV *//
18         SendPacket(PacketType, &Slots);
19 else
20     if Slots > 0 then
21         //* Wait until channel is free *//
22         Slots − −;
```

The MUX then inspects the data in the audio and video queues and decides on the type of packet to send based on the available multimedia streams and force transmission state (lines 7 and 14). Then, it prepares the packet by merging the streams using the protocol structure given in Table 5.1. Finally, the multiplexed stream is sent in a single packet, and the number of used *Slots* may be updated when the multiplexed data size is smaller than the available rate resource (in Algorithm 3, the "&" operator shows that the function can modify data).

To clarify how the scheme works, in Fig. 5.5, a step-by-step run of the multiplexing algorithm is explained for the buffer case given in Fig. 5.1. This example is given for $T = 5$ ms; however, the same method can be applied for different buffer sizes. For the sake of simplicity and without loss of generality, we assume that audio and video information is always available for transmission. However, the same strategy applies to all cases of data availability. When both audio and video data are available at the MUX buffers, the MUX fairly distributes the resources based on the encoder rate settings. This decision is made with a weighting function as follows:

$$w_V = \frac{R_V}{R_A + R_V} \tag{5.1}$$

where R_A and R_V are the constant bitrate settings for the audio and video encoders, respectively, and w_V is the percentage of resources reserved for video when both audio and video

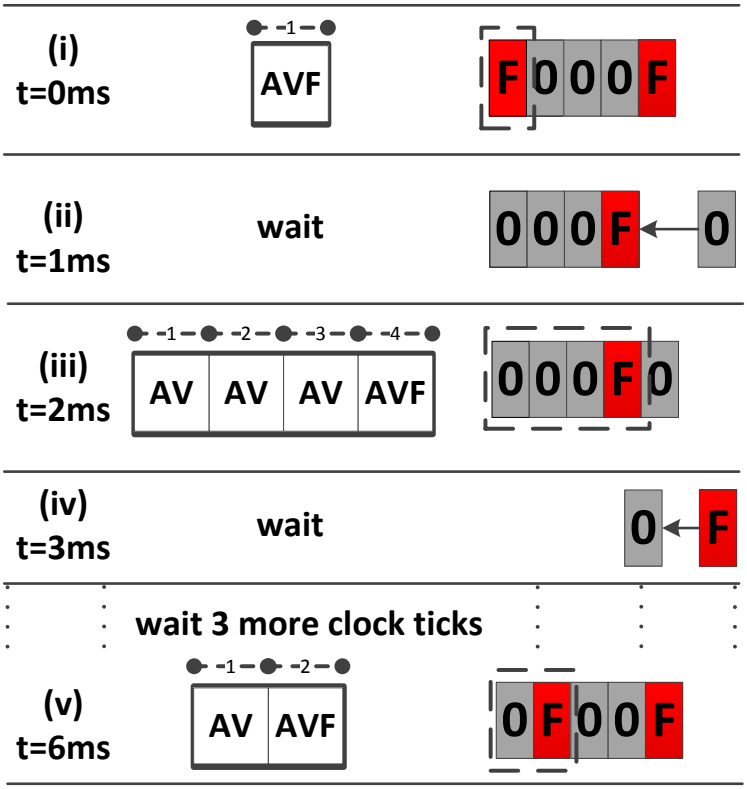


Figure 5.5: Illustration of the multiplexer operation.

data demand transmission resources. In Fig. 5.5, the rectangular blocks on the right are used to show the current state of the force buffer. As shown in Fig. 5.1, the head of the force buffer is on the left, and the tail is on the right. In Fig. 5.5, each square block on the right represents a channel resource bucket for a 1 ms time period. For illustration purposes, we assume a 1 $Mbps$ CBR link, which has 1 ms transmission segments with a length of 1000 $bits$. In the following, we demonstrate how the multiplexing works clock tick by clock tick:

- **(i) t=0ms:** At this instant, we assume that the channel is free to transmit data, which allows the MUX to push new data into the transmission link. The MUX checks the force buffer, starting from the head until reaching a force sample in the buffer. In this example, the MUX encounters the force sample with transmit flag "F" at the head of the force buffer. This force sample is already delayed 4 ms by the buffer, which means that we should not allow a more than 1 ms delay to satisfy the maximum force delay constraint of 5 ms. To achieve this, we can only send 1000 $bits$, which can carry audio, video and force data in one packet. An audio-video-force (AVF) packet occupying 1000

bits, including all header and data bytes, is sent to the OP.

- **(ii) t=1ms:** The MUX waits for the transmission of the previous packet, and a new force sample arrives at the tail of the force buffer. The "Haptic Data Reduction Block" has identified that this new force sample is below the perceptual thresholds when compared to the preceding sample. Hence, the preceding force sample is going to be replicated at the OP side as the representative stimuli of the haptic interaction.

- **(iii) t=2ms:** Now, the channel is free to transmit a new packet, and the MUX checks the sample buffer for free transmission slots. At this time, 4 resource buckets are available to form an audio-video-force (AVF) packet fitting into $1000 \times 4 = 4000$ *bits*. Once this packet is also pushed into the transmission channel, the MUX waits 4 clock ticks before the next packet is transmitted. In this case, the force sample is already delayed by 1 *ms* by the force buffer, and the packet transmission takes an additional 4 *ms* until it reaches the OP side. In total, the force sample is subjected to a 5 *ms* delay, which also hits the target delay constraint on the force signal.

- **(iv) t=3ms:** As seen in Fig. 5.5, the force buffer is filled with a new transmission state "F", and the system continues sleeping 3 *ms* more until the channel becomes free again (see Algorithm 3, lines 20-22).

- **(v) t=6ms:** At this instant, the channel is available again for a new transmission. During the last 3 *ms* of the sleeping period, the force buffer is filled up with "00F" sample sequence, as shown in Fig. 5.5 at $t = 6ms$. For this buffer state, the MUX finds the first force sample with transmit flag "F" in the second slot. This force sample is already delayed 3 *ms* by the buffer. Therefore, an AVF packet fitting into 2 resource buckets ($1000 \times 2 = 2000$ *bits*) is formed and pushed into the transmission channel. As previously shown, the force sample inside the packet is exposed to a 2 *ms* delay in the channel. Hence, the 5 *ms* delay constraint is achieved.

In the following, we present additional packet-merging examples shown in Fig. 5.6 for different buffer cases with the same transmission rate assumption and buffer length.

- **Case (a):** If no important force sample has been identified for the past 5 *ms*, the available transmission rate is allocated completely to the audio and video data waiting in the queues. According to the given channel rate, the allocated resource for video and audio is $1000 \times 5 = 5000$ *bits*, and an AV packet occupying 5 resource buckets is pushed into the channel. Because the available transmission rate is constant (1 *Mbps*), this packet will block the transmission line for 5 *ms*. During this busy time, the MUX does not schedule a new transmission and waits until the current transmission is completed.

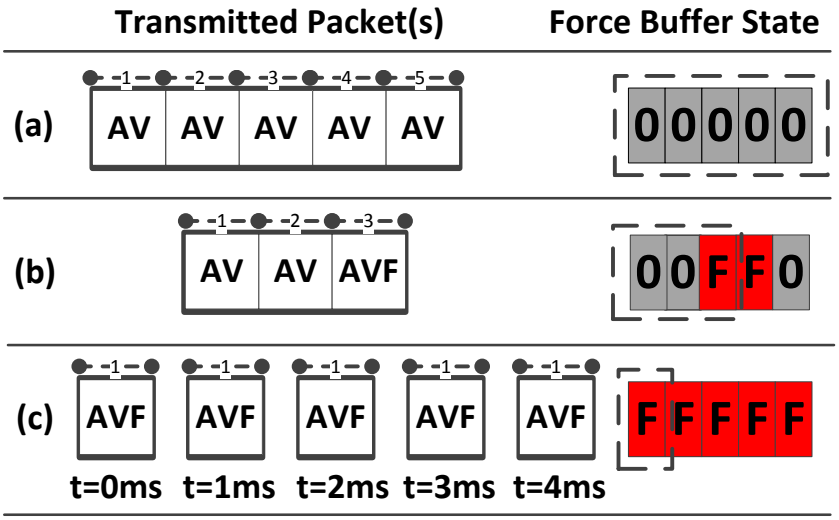


Figure 5.6: Examples for packet generation for different force buffer states.

With this approach, we guarantee a non-busy channel if a force packet arrives after this period.

- **Case (b):** In this example, the MUX encounters the first force transmission at the third slot. Until now, the force sample has been waiting to be transmitted for 2 ms, and we are allowed to delay the sample for a further 3 ms to keep the delay at 5 ms. In this situation, the MUX decides to form an AVF packet using 3 transmission slots ($1000 \times 3 = 3000$ $bits$). We expect that this packet is transmitted in 3 ms. Hence, the 5 ms delay constraint of the force sample is not violated. The fourth transmission slot including "F" is not combined into the packet because the AVF packet occupying 4 transmission slots would introduce a 6 ms delay for the preceding force sample in the third slot, which would violate the delay constraint of 5 ms.

- **Case (c):** This example shows the most critical transmission event, in which the haptic data reduction scheme marks all the samples for transmission in the force buffer. Under this condition, the MUX generates single packets including audio, video and force data and sends them separately at consecutive clock ticks, as illustrated in Fig. 5.6. Using this strategy, the packets do not block each other during transmission. Accordingly, the transmission delay of the adjacent force updates is kept at 1 ms, and the 5 ms delay constraint on the force samples is not violated.

5.2 Demultiplexing

The DEMUX block in Fig. 5.1 performs the parsing of media streams and forwards them to the corresponding decoding and display processing unit. Algorithm 4 gives the demultiplexing procedure for the packets. When a packet is received at the DEMUX side, the MUX header is first parsed to identify the media content of the packet, as shown in line 2. In line 3, the time stamp of the packet is read and written into the acknowledgment packet (see line 5). Concurrently, the acknowledgment packet is prepared to update the delay measurements at the MUX side.

Parse force: Between lines 6 and 12, the force information is extracted if the packet contains force samples. If the packet includes force data, force and energy signals are parsed using the packet structures in Table 5.1, and they are immediately passed to the "Haptic Data Reconstruction" block in Fig. 5.1.

Parse audio: Between lines 13 and 18, the audio data are parsed if the packet includes audio frames. Audio frames, along with their time stamps, are extracted from the packet and passed to the CELT audio decoding block, as shown in Fig. 5.1.

Parse video: Between lines 19 and 29, the video data are demultiplexed if video data are available inside the packet. Video fragments are buffered in the order of their fragment and frame numbers. These numbers help the DEMUX to not overlap consecutive video frames in cases where the last fragment of a frame is lost during the transmission. Due to the real-time constraints, it is not possible to acknowledge the missing parts and retransmit them. In the case of packet losses, the DEMUX continues buffering the video stream and pushes the corrupted video frame to the decoder if the next video frame starts arriving. The multiplexing scheme relies on the video decoder to recover the missing parts of the video frame using its built-in error concealment features.

Finally, the acknowledgment packet is formed and sent to the MUX side. The acknowledgment also has a similar header and packet structure as the MUX packets. Table 5.2 shows an acknowledgment packet for an AVF packet with a similar structure as Table 5.1. The insertion of video and audio time stamps is signaled using bit flags 6 and 7 of the acknowledgment header, as shown in Fig. 5.7. At the MUX side, using the header information, the corresponding delay measurements are computed for transmission rate estimation and congestion control. Similarly, an acknowledgment packet is generated for other types of received packets given in Table 5.1

Table 5.2: Acknowledgment packet structure for a AVF packet.

Acknowledgment packet structure
$H(1) - PTS(2) - SID(2) - STS(2) - NAF(1) - \alpha \times (ATS(2)) - FN(2) - FTS(2)$

Bit	0	1	2	3	4	5	6	7
Header	M			N	N	N	V	A

Figure 5.7: Acknowledgment header (1 byte) structure in bits.

Algorithm 4: Demultiplexing function.

```
if PacketArrives == true then
    PacketHeader = ReadHeader(Packet);
    PacektTimeStamp = ReadTimestamp(Packet);
    //* Add the acknowledgment items *//
    AckPacket = [PacketHeader, PacketTimeStamp];
    if isForceExist(PacketHeader) == true then
        SampleID = ReadID(Packet);
        SampleTimeStamp = ReadTimestamp(Packet);
        Force3DoF = ReadForce(Packet);
        Energy3DoF = ReadEnergy(Packet);
        //* Add the acknowledgment items *//
        AckPacket = [AckPacket, SampleID, SampleTimeStamp];
    if isAudioExist(PacketHeader) == true then
        NumAudioFrames = ReadNAF(Packet);
        [AudioBuffer, AudioFrameNRs, AudioFrameDelays] =
          ReadAudioFrames(Packet, NumAudioFrames);
        AudioDecodeDisplay(AudioBuffer);
        //* Add the acknowledgment items *//
        AckPacket = [AckPacket, AudioFrameNRs, AudioFrameDelays];
    if isVideoExist(PacketHeader) == true then
        FrameNR = ReadFrameNumber(Packet);
        FrameFGN = ReadFragmentNumber(Packet);
        FrameTimestamp = ReadTimestamp(Packet);
        PayloadLength = ReadPLL(Packet);
        VideoBuffer = ReadPayload(Packet);
        if isLastVideoFragment(PacketHeader) then
            VideoDecodeDisplay(VideoBuffer);
            FrameDelay = CurrentTime − FrameTimestamp;
            //* Add the acknowledgment items *//
            AckPacket = [AckPacket, FrameNR, FrameDelay];
    //* Send Acknowledgment *//
    SendAcknowledgment(AckPacket);
```

5.3 Real-time transmission rate estimation and adaptation of system parameters

When we consider a teleoperation session running over the internet, which is shared with other users, or a wireless link with time-varying transmission properties, a decrease in the transmission rate increases the delay and jitter of the signals. Hence, sudden transmission rate drops may cause dangerous situations during the manipulation. To ensure that the system remains on alert for such cases, the available transmission rate of the communication link needs to be instantly tracked, and the throughput of the system and bitrate setting of the video encoder must be adapted accordingly. This section of the chapter closes the loop between the MUX and the DEMUX to predict the available transmission rate of the forward channel (from the MUX to the DEMUX). Therefore, it is possible to adapt the throughput of the MUX, force buffer size and video encoder bitrate for the efficient and low-delay transmission of multiple modalities. First, we address the challenges and methods of transmission rate estimation for a real-time interactive application, and then, we describe the transmission rate estimation algorithm developed in this thesis. Finally, an adaptation scheme for sudden congestion or transmission rate drops is introduced.

5.3.1 Transmission rate estimation

In the literature, several transmission rate estimation schemes [BP95, SSZ98, CGM+02] have been proposed for congestion control, mainly focusing on TCP/IP-based applications. The transmission rate estimation is performed using the time stamps, the delay introduced by the two-way signal propagation which is also called the round trip time ($RTT = t_{prop}^{forward} + t_{prop}^{backward}$, and RTT is used as an abbreviation to represent this delay in the thesis), and the length of the transmitted data packets. In the following, we give the computation of a single sample transmission rate:

$$sampleTR(i) = \frac{BytesReceived(i)}{CurrentTime - PacketTimestamp(i) - RTT} \tag{5.2}$$

where $sampleTR(i)$ is the measured transmission rate from a single packet, $CurrentTime$ is the updated clock tick, $PacketTimestamp(i)$ is the recorded clock tick when the packet is injected into the communication pipeline, and RTT is the delay caused by the signal propagation over the transmission path, which is assumed to be a known/pre-estimated constant and not rapidly changing over time. Although it appears simple and straightforward to compute the true capacity of the communication link from Eq. 5.2, there exist numerical computation issues that drastically degrade the estimation accuracy as follows:

- **Clock frequency:** The sampling frequency of the clock has a strong impact on the transmission rate estimation because it determines the precision of time stamps in the

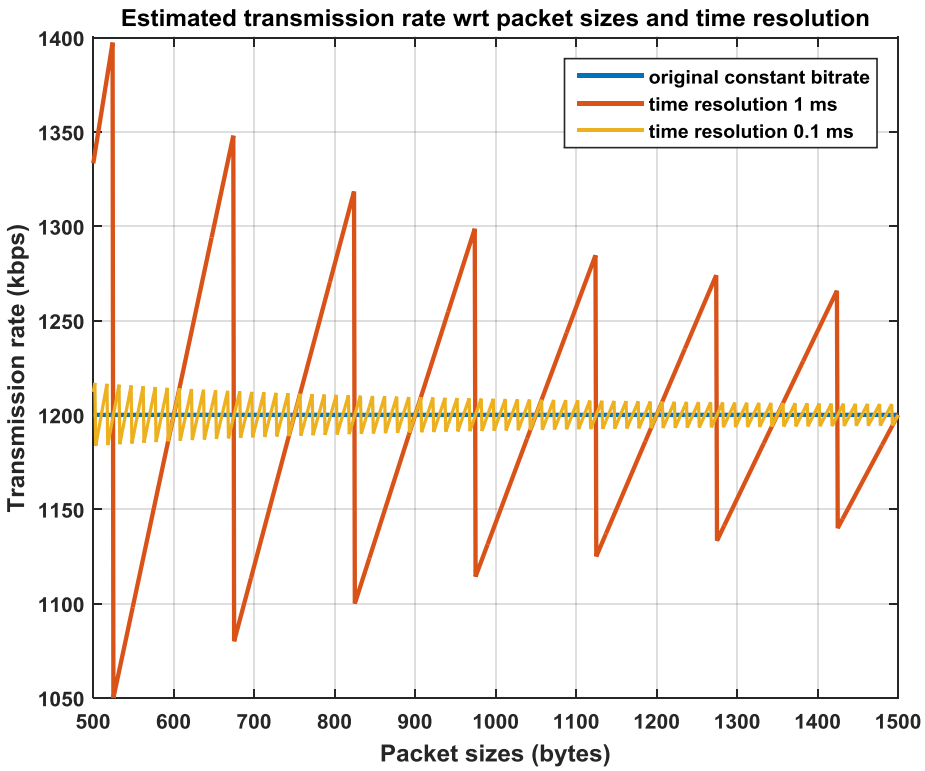


Figure 5.8: Transmission rate measurement using Eq. 5.2 for packet sizes from 500 to 1500 *bytes* and clock resolutions of 1 *ms* and 0.1 *ms*.

denominator of Eq. 5.2. As the probed transmission rate value increases, the estimation error rises as well due to the low clock frequency. Therefore, the accuracy of the transmission rate measurements improves as the clock frequency increases.

- **Packet length:** The packet length also affects the precision of transmission rate measurements in Eq. 5.2. As the size of a packet increases, the accuracy of the measurement improves because it leads to a reduced prediction error due to precision loss of the clock frequency.

To better illustrate the above-mentioned issues, we present the transmission rate measurements for 1 and 10 kHz clock frequencies for packet lengths from 500 to 1500 *bytes* in Fig. 5.8. In this example, the transmission rate is set to a constant bitrate of 1.2 $Mbps$, and applying Eq. 5.2, the transmission rate measurement for each packet length is computed using the time stamps sampled with 1 and 10 kHz clocks. As shown in Fig. 5.8, sampling the time stamps at 10 kHz yields better measurements close to the original bitrate, and the measurement error can be reduced if longer packet lengths are employed in the estimation.

It is important to note that the usage of Eq. 5.2 will theoretically yield a biased estimation, and it is not easy to filter the bias by employing even sophisticated filters [CFM04]. The bias of the estimation is briefly described in [CFM04]. In the following, we give a detailed derivation of the bias term when the measurements of Eq. 5.2 are employed in transmission rate estimation.

Let S and T be the random variables representing the packet length and transmission time of the corresponding packet. If we consider Eq. 5.2 as a candidate estimator $\hat{B}$, then the random variables generated by this expression can be computed as $\hat{B} = \frac{S}{T}$. To simplify the derivations, S and T are considered as statistically independent random variables [CFM04]. The mean of $\hat{B}$ can be determined as follows:

$$E[\hat{B}] = E[S] \cdot E[\frac{1}{T}] \tag{5.3}$$

where $g(T) = \frac{1}{T}$ is a function of the random variable T. If we apply a Taylor series expansion of $g(T)$ at the point $\mu_T = E[T]$, then the following expression is obtained:

$$\begin{aligned} g(T) &= \sum_{n=0}^{+\infty} \frac{g^{(n)}(\mu_T)}{n!}(T-\mu_T)^n \\ &= \frac{1}{\mu_T} - \frac{1}{(\mu_T)^2}(T-\mu_T) + \frac{1}{(\mu_T)^3}(T-\mu_T)^2 + ... \\ &= \sum_{n=0}^{+\infty}(-1)^n \frac{(T-\mu_T)^n}{(\mu_T)^{n+1}} \end{aligned} \tag{5.4}$$

To obtain the term $E[\frac{1}{T}]$, we apply the linear expectation operator to the final expression in Eq. 5.4 and obtain the following:

$$E[g(T)] = \sum_{n=0}^{+\infty}(-1)^n \frac{E[(T-\mu_T)^n]}{(\mu_T)^{n+1}} \tag{5.5}$$

When we substitute Eq. 5.5 into Eq. 5.3, we obtain the following expression for the mean of the estimator $\hat{B}$:

$$\begin{aligned} E[\hat{B}] &= E[S] \cdot \sum_{n=0}^{+\infty}(-1)^n \frac{E[(T-\mu_T)^n]}{(\mu_T)^{n+1}} \\ &= \frac{E[S]}{E[T]} + \sum_{n=1}^{+\infty}(-1)^n \frac{E[(T-\mu_T)^n]}{(\mu_T)^{n+1}} \end{aligned} \tag{5.6}$$

In Eq. 5.6, the first term $\frac{E[S]}{E[T]}$ represents the true average transmission rate $E[B]$ used by the packets, and the second term represents the bias term coming from the statistics of the transmission time T.

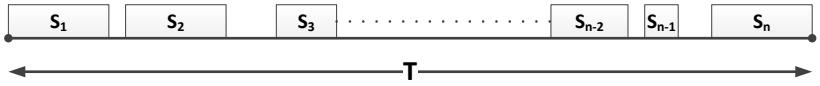


Figure 5.9: TIBET transmission rate estimation packet pattern (reproduced from [CFM04]).

To avoid the bias effect on the estimation of the true transmission rate, in [CFM04], the authors proposed a non-linear filtering method called the Time-Intervals-based Bandwidth Estimation Technique (TIBET), which is an improved version of TCP Westwood algorithms [CGM+02]. TIBET cleverly focuses on the first term in Eq. 5.6 to achieve an unbiased solution for the transmission rate estimation problem. As understood from its name, this technique analyzes the packet lengths and transmission time based on a windowing approach.

In Fig. 5.9, we illustrate the transmission of n packets within a time window of T, and S_i represents the length of each packet. The average transmission rate in the time interval T, $E[B]$, can be computed as follows:

$$E[B] = \frac{1}{T}\sum_{i=1}^{n} S_i \tag{5.7}$$

Inspired by the first term of Eq. 5.6, the scheme applies separate estimators $\hat{\mathbf{S}}$ and $\hat{\mathbf{T}}$ for the average packet length $E[S]$ and average transmission time $E[T]$. Therefore, it is possible to form a combined estimator for the transmission rate as follows:

$$\hat{\mathbf{B}} = \frac{\hat{\mathbf{S}}}{\hat{\mathbf{T}}} = \frac{E[S]}{E[T]} \tag{5.8}$$

As ensured from Eq. 5.7, the estimator $\hat{\mathbf{B}}$ is expected to satisfy the constraint of an unbiased estimator: $E[\hat{\mathbf{B}}] = E[B]$. In [CFM04], the average packet length and transmission time estimators $\hat{\mathbf{S}}$ and $\hat{\mathbf{T}}$ are implemented as first-order low-pass IIR filters in the following expressions:

$$\begin{aligned} \hat{\mathbf{S}} &: AvgPacLen(i) = \alpha \times AvgPacLen(i-1) + (1-\alpha) \times CurrSampLen(i) \\ \hat{\mathbf{T}} &: AvgSampTime(i) = \alpha \times AvgSampTime(i-1) + (1-\alpha) \times CurrSampTime(i) \end{aligned} \tag{5.9}$$

where α is the pole of the low-pass filters, which is between $0 \leq \alpha \leq 1.0$ and has a critical effect on the estimation performance. As α decreases, the algorithm becomes highly sensitive to changes in the available transmission rate. However, this causes oscillations in the estimation. In contrast, as α approaches 1, the algorithm produces stable estimates but becomes less sensitive to changes in the network. For the teleoperation streams, it is experimentally found that 0.995 represents a compromise for achieving a balancing between a response to changes in the transmission rate and estimation accuracy.

The average transmission rate at time instant i, $TR_{avg}(i)$, is estimated as follows:

$$\hat{\mathbf{B}} : TR_{avg}(i) = \frac{AvgPacLen(i)}{AvgSampTime(i)} \tag{5.10}$$

However, the oscillations of the $TR_{avg}(i)$ measurements are still very high and require further processing to reach a smooth estimate of the transmission rate. Therefore, a non-linear filtering method called the Core Stateless Fair Queuing (CSFQ) estimation algorithm [SSZ98], which has been developed for the estimation of the transmission rate between IPv4-based routers, is employed to smooth out the oscillations in Eq. 5.10 as follows:

$$TR_{est}(i) = (1 - e^{-\frac{T_{est}(i)}{T_{const}}}) \cdot TR_{avg}(i) + e^{-\frac{T_{est}(i)}{T_{const}}} \cdot TR_{est}(i-1) \tag{5.11}$$

The adaptive filter coefficient is an exponential function that adjusts the weight of the estimation based on the time interval between adjacent estimations as follows:

$$T_{est}(i) = CurrentTime - LastEstimationTime \tag{5.12}$$

If the time difference between two estimations increases, the filter relies more on the current measurement. In particular, during congestion events, the recent measurements give reliable estimations; therefore, they should be highly weighted. This is provided by the adaptive filter weight with the increasing estimation intervals, $T_{est}(i)$, due to congestion. In contrast, the filter increases the transmission rate conservatively when the transmission capacity improves. In Eq. 5.11, T_{const} is a time constant, and it is recommended to be set in the range between 0.1 and 0.5 seconds [SSZ98]. Finally, $TR_{est}(i)$ is used as the final stable estimate of the transmission rate. In Fig. 5.10, we illustrate how the TIBET approach is employed in the teleoperation system. The communication channel from the MUX to the DEMUX shown inside the dashed rectangle is subject to the estimation, and the two-way signal propagation delay (RTT) is known or pre-estimated. In such a real-time system, the time stamps are sampled according to the loop rates, which depend on the computation speed of the loop and thread scheduling of the operating system. Hence, it is mandatory to make the implementation on a real-time operating system, which guarantees a firm scheduling period for each thread. In Fig. 5.10, the processing blocks, "MUX", "DEMUX" and "TIBET", run as threads with 1 and 10 kHz loop rates as shown. The "MUX" block works in sync with the haptic sampling loop to schedule the audio, video and force signals as described in Section 5.1.2, and the block reads the current transmission rate and time stamps of each packet from the "TIBET" block. The "DEMUX" block listens to the communication channel every 100 μsec and transmits the packet time stamp (PTS) back to the "TIBET" block over the backward channel. Similarly, the "TIBET" block listens for incoming acknowledgment packets every 100 μsec, performs the described transmission rate estimation technique, TIBET, and updates the "MUX" block to the current transmission rate as long as it receives an acknowledgment packet. It is important to mention that the time measurements can be affected by the processing time of

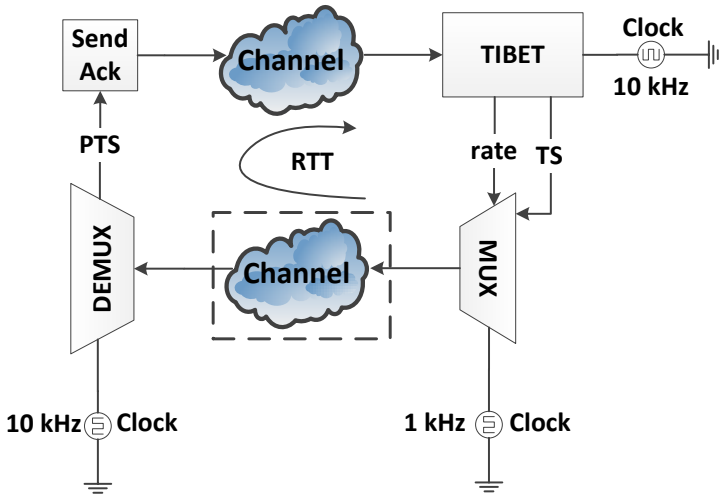


Figure 5.10: Transmission rate estimation loop rates.

each thread, which may lead to a slight underestimation of the transmission rate. To speed up the computations of TIBET, fixed point arithmetic is used for the implementation of IIR filters, and accelerated assembly equivalents of math and memory functions are employed. In Algorithm 5, we give a summary of the transmission rate estimation thread.

Algorithm 5: Summary of the transmission rate estimation algorithm, TIBET.

if $AckReceived == true$ **then**
  //* **Convert received bytes into bits** *//
  $CurrSampLen(i) = BytesReceived(i) \times 8$;
  //* **The time spent in the channel is computed** *//
  $CurrSampTime(i) = CurrentTime - PacketTimestamp(i) - RTT$;
  //* **The IIR filters in Eq. 5.9** *//
  $AvgPacLen(i) = \alpha \times AvgPacLen(i-1) + (1-\alpha) \times CurrSampLen(i)$;
  $AvgSampTime(i) = \alpha \times AvgSampTime(i-1) + (1-\alpha) \times CurrSampTime(i)$;
  //* **The final non-linear filtering in Eq. 5.11** *//
  $T_{est}(i) = CurrentTime - LastEstimationTime$;
  $TR_{est}(i) = (1 - \exp(\frac{-T_{est}(i)}{T_{const}})) \times \frac{AvgPacLen(i)}{AvgSampTime(i)} + \exp(\frac{-T_{est}(i)}{T_{const}}) \times TR_{est}(i-1)$;
  //* **Update parameters for next iteration** *//
  $LastEstimationTime = CurrentTime$;
  $AvgPacLen(i-1) = AvgPacLen(i)$;
  $TR_{est}(i-1) = TR_{est}(i)$;
  $AvgSampTime(i-1) = AvgSampTime(i)$;
  $AckReceived = false$;
  $NewTREstimation = true$;

5.3.2 Bitrate adaptation

Whenever a new transmission rate estimate is made and passed to the MUX, the bitrate adaptation algorithm updates the multiplexing throughput rate immediately. Then, the multiplexing buffer size is updated in line with the current transmission rate and MTU size of the communication medium. The MUX needs to limit the packet lengths to the MTU size (1500 *bytes* if UDP/IPv4 over ethernet is used) of the transmission protocol because packets larger than the MTU size will be fragmented by the ethernet protocol, leading to an increase in the packet rates and usage of extra fragmentation headers over the network. In case the packets are fragmented at the data link layer and one of the fragments is lost during the transmission, it is not possible to recover the packet using the remaining fragments. Therefore, it is important to fix the size of a packet to the MTU size at the application layer. This constrains the MUX to reschedule if the size of the packet being prepared reaches the MTU size. Considering this limitation, the maximum multiplexing buffer size is bounded analytically by the MTU

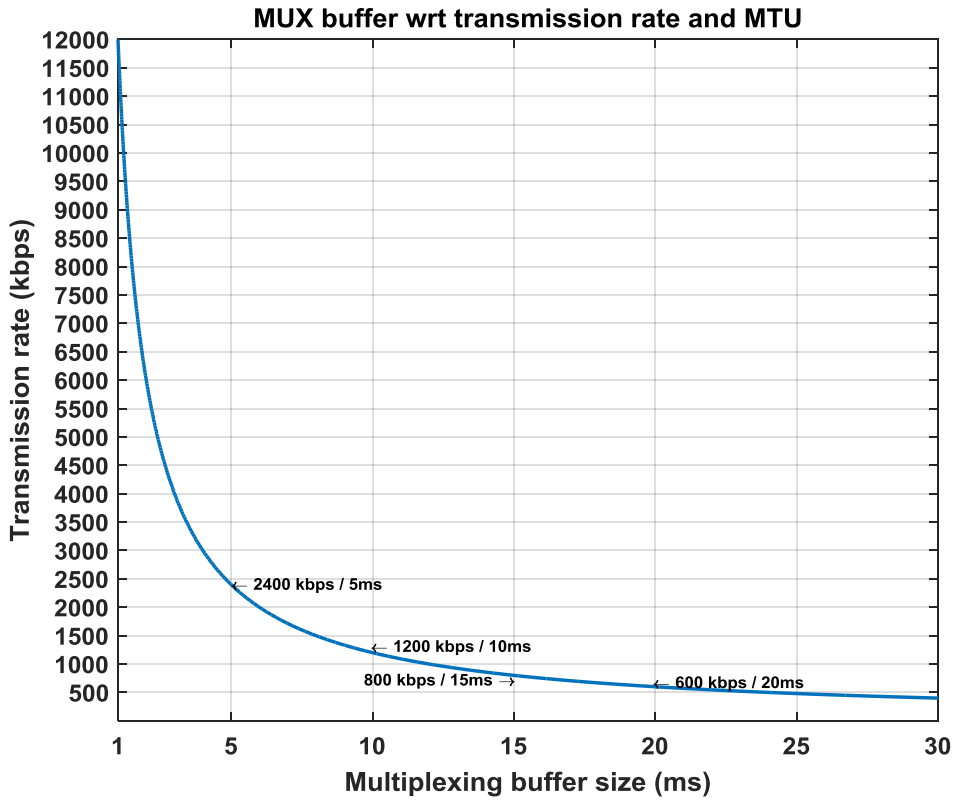


Figure 5.11: MUX buffer size with respect to transmission rate and fixed MTU size (1500 *bytes*).

size and the current transmission rate estimate, as shown in the following Eq. 5.13:

$$BufferSize = \frac{MTUsize \times 8(bits)}{TR(kbits/sec)} \tag{5.13}$$

In Fig. 5.11, we plot the given relation in Eq. 5.13 for transmission rate and buffer size by considering a fixed MTU size of 1500 *bytes*. For example, the MUX does not need to buffer more than 5 force samples under a transmission capacity of 2.4 *Mbps* because 5 resource slots are sufficient to fill a packet given the MTU size according to the merging decisions given in Section 5.1. On the other hand, it is also possible to set the buffer size lower than the bound; however, this selection will yield a higher packet rate and greater header usage accordingly. In case the buffer size is higher than the bound, the MUX does not allow a packet to have a size that is larger than the MTU. The MUX forms a packet that fits into the MTU and then reschedules the remaining data after the transmission of the current packet. Therefore, the bitrate adaptation stage needs to set the multiplexing buffer size to the maximum boundary to produce the minimum possible packet rate to achieve efficient usage of the available transmission rate. However, from the implementation perspective, it is difficult to update the multiplexing buffer size instantly because the buffer is continuously used as a queue in the loop, and frequent updates may cause jitter of the force signal. Thus, the MUX updates the buffer size in multiples of 5, given the loop interrupt issues and frequent queue manipulation operations. Additionally, the buffer size is set to its ceil value to guarantee the full utilization of resources at all times. The math operations on the buffer size computation are as follows:

$$\begin{aligned} OriginalBufferSize &= round(\frac{MTUsize \times 8}{MultiplexerThroughput}) \\ NewBufferSize &= ceil(\frac{OriginalBufferSize}{5}) \times 5 \end{aligned} \tag{5.14}$$

Video bitrate adaptation

As discussed in Section 3.2, the video frame rate, f, is also the arrival rate of the video frames. To avoid the queuing delay of the video, each block processing video frames has to finish its task in less than the video arrival period, $\frac{1}{f}$. For instance, the task completion deadline for a 25 *fps* video is 40 *ms* for each video processing unit. The MUX and communication link are the last stages that affect the video delay. The MUX has to pass the current video frame before the next frame arrives at its input so that the video frames are not queued at the MUX. Therefore, the service time of the communication link needs to be not only less than the frame arrival period but also close to it for the utilization of the available transmission rate and achievement of good visual quality. Considering the MUX block in the system as a bitrate shaper, we can superimpose their delay effect and determine the available transmission rate at the video input. The goal is to design a single-frame delay constraint scheme that can

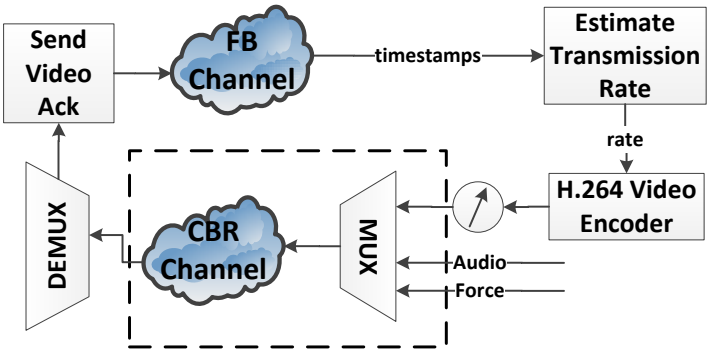


Figure 5.12: Video bitrate estimation setup.

guarantee a target delay for each video frame. Inspired by [DvBA08], a single-frame delay constraint can be achieved as follows:

$$\frac{FrameSize}{VidTR} \leq \Omega \cdot DelayConstraint \tag{5.15}$$

where Ω is a safety factor, $0 < \Omega \leq 1$, against transmission rate and backlog estimation errors. Using the relation in Eq. 5.15, the corresponding frame size demand is sent to the video encoder to achieve the target $DelayConstraint$. Therefore, we need to determine the maximum achievable transmission rate at the video input of the MUX to obtain the correct setting of $FrameSize$. In Fig. 5.12, we propose a method to build a transmission rate model that represents the combined delay effect of the MUX and the channel. In this setup, we run the system over known CBR channels with transmission rates from 800 to 2100 *kbps* in steps of 100 *kbps* using pre-recorded teleoperation sessions. As observed in Fig. 5.12, we consider the channel and the MUX in the dashed box as a combined network bottleneck, and the audio and force signals are the incoming side traffic. We apply transmission rate estimation at the video input of the MUX using Algorithm 5. For the transmission rate estimation, we treat every video frame as a single packet, and the DEMUX sends an acknowledgment packet when the transmission of a frame is completed. The estimated transmission rate for each bitrate condition is recorded and averaged. In Fig. 5.13, dots indicate the relation between the transmission capacity of the channel and the estimated average video bitrate that can be pushed into the system. As observed from the figure, there is a linear relationship between the channel capacity and the video bitrate. A linear model is fitted and used in combination with the single-frame delay constraint given in Eq. 5.15 for the transmission rate adaptation as follows:

$$VideoBitrate = (\beta \times TR_{est}(i) - S) \times VideoDelayConstraint \times framerate \tag{5.16}$$

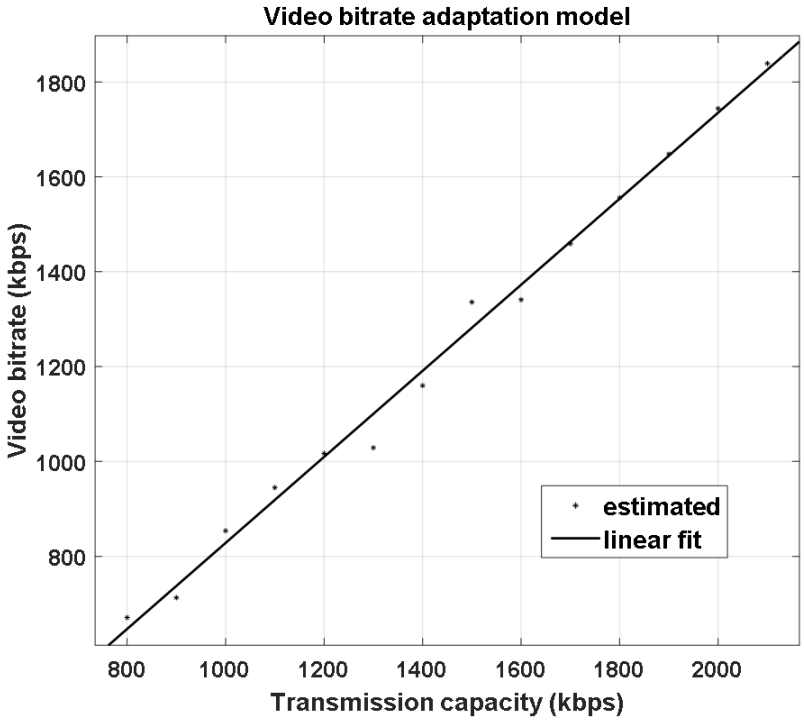


Figure 5.13: Bitrate model for transmission capacities from 800-2100 *kbps*.

According to the linear fit, the model parameters β and S are found to be 0.87 and 89 *kbps*, respectively. Using the delay constraint, the channel and MUX together finish serving each frame before the successive frame arrives. Finally, the resource allocation weighting factor calculation in Eq. 5.1 between the audio and video is updated using the new bitrate setting of the video stream. In the following Algorithm 6, we summarize the bitrate adaptation described in this section.

Algorithm 6: Summary of the bitrate adaptation algorithm.

if $NewTREstimation == true$ **then**
  //* **Update the MUX throughput rate** *//
  $MultiplexerThroughput = TR_{est}(i)$;
  //* **Buffer size updates in** 5 ms **steps to avoid frequent changes** *//
  $OriginalBufferSize = round(\frac{MTUsize \times 8}{MultiplexerThroughput})$;
  $NewBufferSize = ceil(\frac{OriginalBufferSize}{5}) \times 5$;
  $ApplyNewMuxBuffer(NewBufferSize)$;
  //* **Compute the new video bitrate using Eq. 5.16** *//
  $VideoBitrate = (\beta \times TR_{est}(i) - S) \times VideoDelayConstraint \times framerate$;
  //* **Update weight for audio-video rate allocation** *//
  $w_v = \frac{VideoBitrate}{VideoBitrate + AudioBitrate}$;
  $NewTREstimation = false$;

5.3.3 Congestion detection and control

The transmission rate estimation algorithms become error prone as the RTT between the sender and the receiver increases. Sudden congestion events are considered as unexpected transmission rate drops due to increased side traffic in the network, which may have dangerous consequences during telemanipulation. As a result of delayed acknowledgment packets, the detection of the transmission rate drops lags behind. Hence, there exists an uncertain period wherein the system disregards the decreasing communication capacity and continues the transmission of the data at high rates. When the true available transmission rate information arrives at the sender side, even though the estimation is correctly performed and the bitrates are updated immediately, the previously transmitted data are going to aggravate the network bottleneck. In this section, we focus on the effects of unexpected transmission rate drops and propose a congestion detection and control scheme for the teleoperation system.

Fig. 5.14 illustrates the system behavior that is recorded during a real teleoperation session

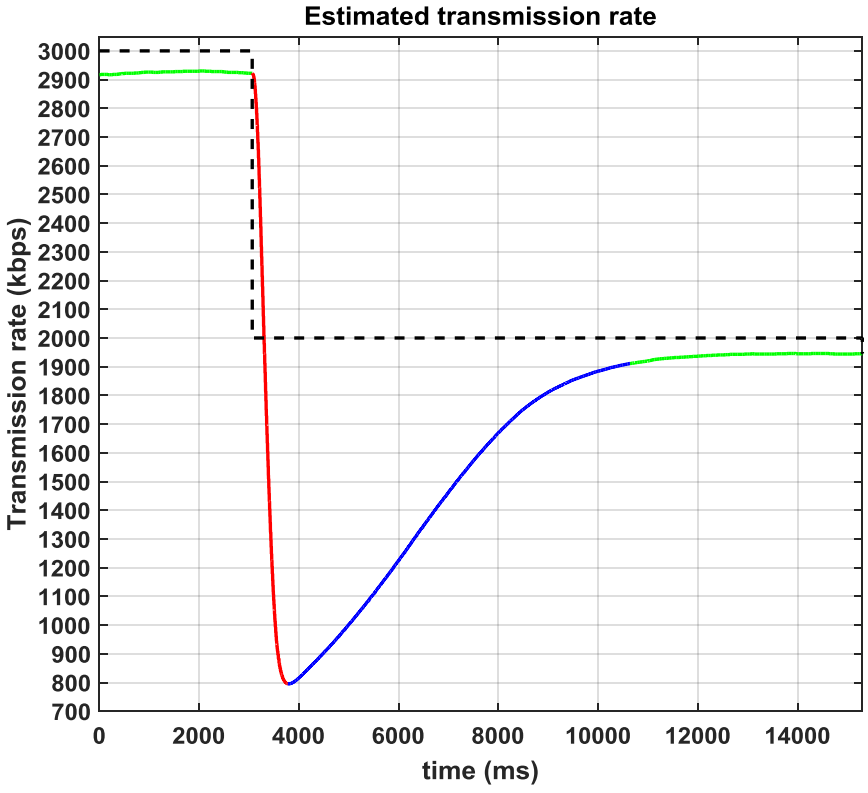


Figure 5.14: Congestion detection processing regions: The dashed line represents the original available transmission rate. At $t = 3064\ ms$, the available transmission rate drops from 3 $Mbps$ to 2 $Mbps$. The colored lines illustrate the behavior of the transmission rate estimation algorithm when the two-way signal propagation delay is $RTT = 100\ ms$.

at the moment of a sudden transmission rate drop. In this example, the network capacity abruptly decreases from 3 $Mbps$ to 2 $Mbps$ (shown as a dashed line) due to the increased traffic in the network, and we show the transmission rate estimation of the system with the colored graph. The green line represents the correct estimation of the available transmission rate and stable regions for teleoperation. The red line refers to the uncertain period, in which the transmission rate estimation fails due to the unexpected congestion. The blue line shows the correction period of the transmission rate estimation algorithm converging to the true available transmission rate.

When congestion occurs, the system enters an uncertain state, and the capacity of the network is unknown in the interim. Consequently, until the transmission rate estimation converges to the true capacity of the link, the current system parameters remain ambiguous, and simultaneously, the delay constraints on the signals are violated. A trivial solution to the problem would be to stop the system immediately and restart it from the lowest possible bitrate setting. However, the transition would interrupt the user's interaction, and it would take time to converge to the true available transmission rate. On the other hand, if congestion occurs very frequently, the stop and adaptation states can become annoying to the OP. Concerning a smooth transition to the new network conditions, an intelligent video frame dropping and bitrate adaptation strategy needs to be applied.

Fig. 5.15 illustrates the challenge of congestion detection due to RTT delays. In this example, the RTT delay is 160 ms, which is symmetrically partitioned as shown in Fig. 5.15. In addition to the RTT delay, the channel transmission adds its own queuing delay based on its current service time for packets. Based on the model that we developed in Section 5.3.2, the service time of a frame is ensured to be slightly less than the frame arrival period of 40 ms. Let us assume that the congestion starts at the transmission of frame 5: Fr_5. In this case, it would take at least 160 ms for the transmission rate estimation block "TIBET" to detect the congestion event. During this time period, the "MUX" block has already pipelined at least 4 frames using the former transmission rate estimations. Once the transmission rate drop starts, the adaptive factor of the estimator begins giving greater weight to the recent measurements, and the system enters the uncertain period shown in Fig. 5.14 as a red line. The drop detection decision relies on multiple conditions as follows:

(i) The derivative of the transmission rate signal is tracked in real time using the following 4^{th}-order FIR filter:

$$TRderivative(i) = \frac{((TR_{est}(i-3) - TR_{est}(i)) + 2 \cdot (TR_{est}(i-2) - TR_{est}(i-1))}{8} \tag{5.17}$$

If the derivative exceeds a predefined threshold ($Thr = 1.5$), a congestion event is possible. An increasing positive derivative of the transmission rate indicates a congestion event with a sudden rate drop, and correspondingly, a decreasing negative derivative shows an increasing transmission rate.

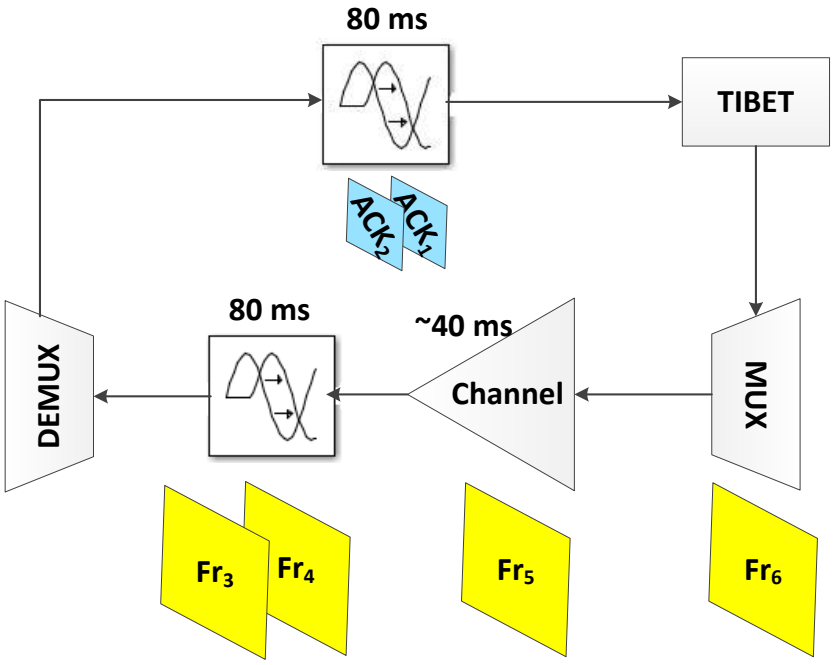


Figure 5.15: The video frames and acknowledgment packets in the network for a symmetric propagation delay of 80 ms in one direction. This figure shows the delayed recognition of congestion events. If congestion starts during the transmission of Fr_5, the TIBET and MUX blocks are unaware of the transmission rate change until receiving the acknowledgment packet of Fr_5. During this period, the MUX block has already pushed packets containing at least 4 video frames using the overestimated transmission rate.

(ii) The delays of each force sample and video frame are fed back to the "MUX" side. If one of the delay constraints is violated, this is a strong sign of a congestion in the network.

The congestion control mode is activated when either the delay constraint on the force samples or that on the video frames is violated, and the derivative of the transmission rate is higher than a threshold. The following actions are performed following congestion detection:

(1) **Red line in Fig. 5.14:** As the congestion begins, the current video transmission buffer is discarded, and the video streaming breaks until all unacknowledged frames are acknowledged. Then, the system continues probing the available transmission rate by restarting the video transmission with half of the frame rate (temporal scalability) and intra-only mode. The system may have already streamed a portion of the current video frame, which is discarded due to congestion. At the receiver side, the video decoder applies error concealment techniques to recover the missing parts of the discarded video frame. However, this approach causes visual artifacts in the motion scenes. On the other hand,

the successive video frame should be an intra frame to avoid error propagation for inter predictions. With this approach, we can safely sample the transmission rate to quickly converge to the true transmission rate, and the inter prediction errors in the video stream are avoided because cleared buffers are quickly recovered with intra frames.

(2) **Blue line in Fig. 5.14:** When the delay constraints return to normal levels and when $TRderivative$ reaches a value lower than the threshold, the system enters the recovery state and retains its low rate state until a waiting time threshold is reached. Then, the system increases the video frame rate back to normal and switches back to the inter-intra coding mode.

In Algorithm 7, we give a summary of our proposed method for congestion detection and control for the TIBET Algorithm 5.

Algorithm 7: Summary of the congestion detection and control scheme.

```
//* Compute the derivative of the TR signal *//
TRderivative(i) = ((TR_est(i − 3) − TR_est(i)) + 2 ∗ (TR_est(i − 2) − TR_est(i − 1)))/8;
//* Check the conditions for a congestion event *//
if TRderivative(i) >= Thr&&(ForceDelay > ForceDelayConst||VideoDelay >
  VideoDelayConst) then
    //* The operations during an uncertain period (red line in Fig. 5.14) *//
    if UnackedVideoNumber > 0&&SignalStop == true then
        EmptyVideoBuffer(); //* clear the video buffer *//
        //* Stop video transmission until all frames are acknowledged *//
        VideoTransFlag = false;
    else
        //* All frames are now acknowledged; start probing the channel with
          low-rate video transmission *//
        FPSscale = 2; //* Stream at half frame rate *//
        VideoMode = IntraOnly; //* Stream intra only video to recover the
          errors *//
        VideoTransFlag = true; //* Start video transmission *//
        SignalStop = false;
else
    //* The operations during the converging back period (blue line in Fig.
      5.14) *//
    if WaitingTime > WaitThr then
        FPSscale = 1; //* Switch back to the full frame rate *//
        VideoMode = InterIntra; //* Change video mode back to IPP *//
        //* Make the signal ready for the next congestion event *//
        SignalStop = true;
```

5.4 Experimental setup and results

This section presents extensive experiments and describes their results. The evaluation of the multimodal multiplexing scheme is conducted using a teleoperation system implemented with the following hardware: a KUKA LWR arm [KUK], a JR3 force/torque sensor [JR383], an Allied Vision Mako GigE camera [All16], a Force Dimension Omega 6 haptic device [For01] and an Apposite Netropy 60 network emulator [App16], which separates the OP and TOP sides physically by introducing network bottlenecks and delay-jitter. Fig. 5.16 shows the experimental setup at the TOP side. The camera is mounted and fixed on the robot hand (eye-on-the-hand) monitoring the single-point metal end-effector, the manipulation platform and objects (see example scenes in Fig. 4.6). The dimensions of the manipulation tool are given in Fig. 5.17(a). Its total length from bottom to tip-end is 165 mm, and the tip diameter is semi-conical from 8.8 to 14 mm to fit into the object holes (13 mm, given in Fig. 5.17(c)) for manipulation tasks involving holding and dragging. Fig. 5.17(b) illustrates the construction of the manipulation platform. As observed from Fig. 5.17(b), the dimensions of the triangle- and square-shaped holes are drilled slightly larger (+5 mm) to perform the pegging operation smoothly.

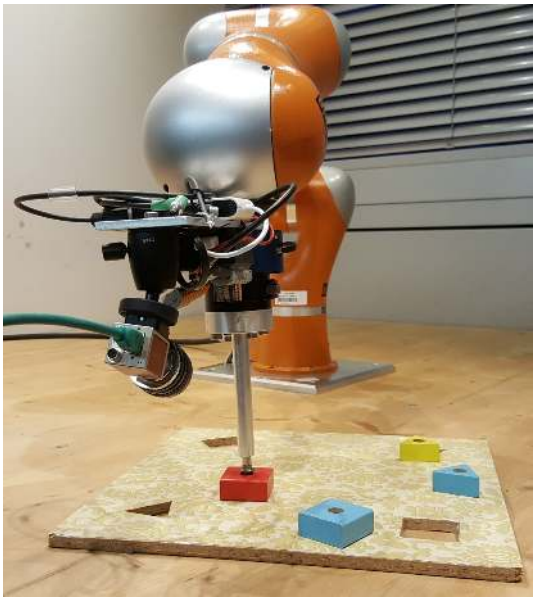
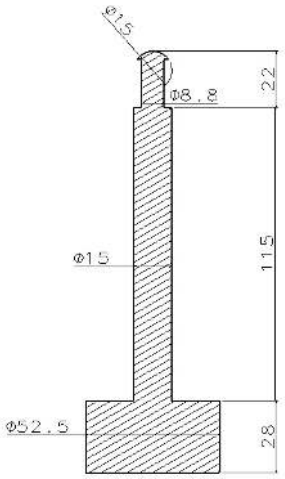
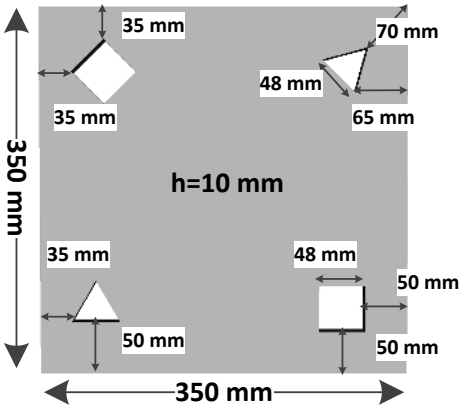


Figure 5.16: Teleoperator KUKA LWR arm with fixed finger manipulator (single-point metal end-effector) performing peg-in-hole task.

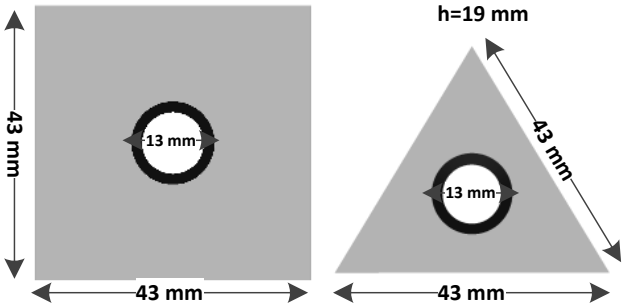
The resolution of the video stream is set to 1280×720 at 25 fps, and the delay constraint of the video stream is adjusted to a fixed value of 35 ms. Initially, the multiplexing buffer is set to 25 ms, and the multiplexing throughput rate is set to 600 $kbps$. The initial video bitrate is computed using Eq. 5.16 with the parameters $\beta = 0.87$, $S = 89$ and $VideoDelayConstraint = 35\ ms$. When the system starts and estimates the available transmission rate, the multiplexing



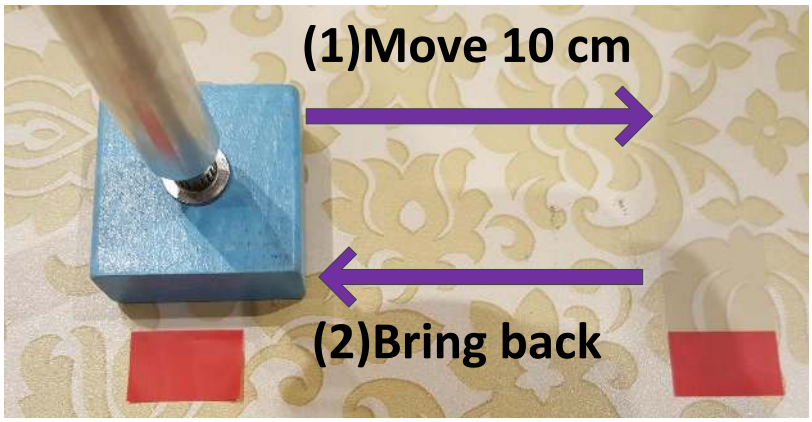
(a) Single-point metal end-effector, dimensions in mm



(b) Manipulation platform dimensions



(c) Dimensions of the objects for manipulation



(d) The procedure of the experiments

Figure 5.17: Construction of the peg-in-hole experiment.

buffer and video bitrate are adapted automatically.

Fig. 5.17(d) illustrates how we performed the experiments 1 and 2. For consistency between different experimental sessions and reproducible results, we force the OP to make a controlled movement during the manipulation. The procedure is as follows:

1. At the beginning the end-effector is in free-space above the object. The OP slightly moves towards the object and gets in contact with it by holding from its hole.

2. The OP drags the object to the target location which is placed 10 cm apart as shown in Fig. 5.17(d).

3. When the target is achieved, the OP releases the object and moves upwards.

4. The OP approaches to the object once again to drag it back to the initial position.

5. Similarly, it holds the object brings it back to the initial position and releases the object.

5.4.1 Experiment 1: Teleoperation over CBR links

In this experiment, we demonstrate the performance of the system when the network transmission rate is constant over time. The tested CBR links are 1, 2 and 3 $Mbps$ without any packet loss and with a symmetric RTT delay of 100 ms. During the teleoperation sessions, the signal delays, the estimated transmission rate, the number of transmitted packets and the PSNR for the visual quality are instantly probed to evaluate the teleoperation system together with the multiplexing scheme.

Fig. 5.18 shows the resulting transmission rate estimation using Algorithm 5. As observed in Fig. 5.18, the transmission rate estimation converges to the true capacity of the links without any overshoot. However, we observe a slight precision loss as the link capacity increases due to the issues discussed in Section 5.3.1. In Table 5.3, we report the precision loss in the transmission rate estimation in the 2^{nd} and 3^{rd} columns as the standard deviation (TR^{σ}_{est})

Table 5.3: This table illustrates the transmission rate estimation performance for the CBR links, packet rate of the system and visual quality of the teleoperation scenes.

TR	**TR_{est} performance**			**Packet rate**		**Visual quality PSNR**			
$(kbps)$	TR^{mean}_{est} $(kbps)$	TR^{σ}_{est} $(kbps)$	TR^{rmse}_{est} $(kbps)$	$mean$ (pac/sec)	$stddev$ (pac/sec)	$mean$ (dB)	$stddev$ (dB)	min (dB)	max (dB)
1000	989.25	2.83	11.11	139.68	6.76	32.40	3.43	28.87	40.82
2000	1964.55	4.10	35.68	192.26	4.11	37.63	3.22	30.44	42.39
3000	2938.22	9.17	62.45	251.09	7.82	40.12	2.84	33.26	43.33

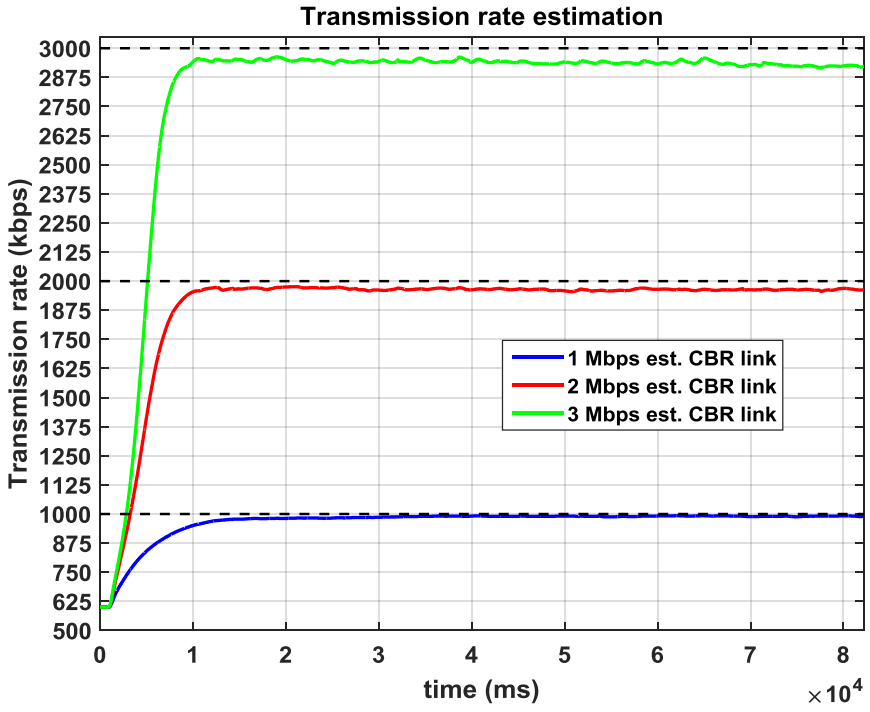


Figure 5.18: Performance of the transmission rate estimation algorithm for CBR links of 1, 2 and 3 $Mbps$.

from the mean estimation (TR_{est}^{mean}) and the root mean squared error (TR_{est}^{rmse}) with respect to the original transmission rate (TR), which are computed as the follows:

$$TR_{est}^{mean} = \frac{1}{N}\sum_{i=1}^{N} TR_{est}(i), \quad TR_{est}^{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(TR_{est}(i) - TR_{est}^{mean})^2} \tag{5.18}$$

$$TR_{est}^{rmse} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(TR - TR_{est}(i))^2} \tag{5.19}$$

However, the mean of the estimation given in the 1^{st} column remains very close to the target bitrate such that, in the tested transmission rate range, the system can approximately estimate $\frac{TR_{est}^{mean}}{TR} \times 100 \approx 97 - 99\%$ of the available transmission rate. If we check the packet rate columns in Table 5.3, the average packet rate increases due to the increased amount of data to be transmitted. Nevertheless, the packet rate results are very promising compared to haptic communication systems without data reduction schemes, which operate at 1000 $packets/second$. Using perceptual haptic data reduction and multiplexing schemes, it remains possible to have a packet rate reduction of approximately $75 - 87\%$ for audio, video and haptics if we take 1000 $packets/second$ as a reference. If we compare this result with systems that transmit only haptic signals, with a data reduction of 90% [HHC+08], we can

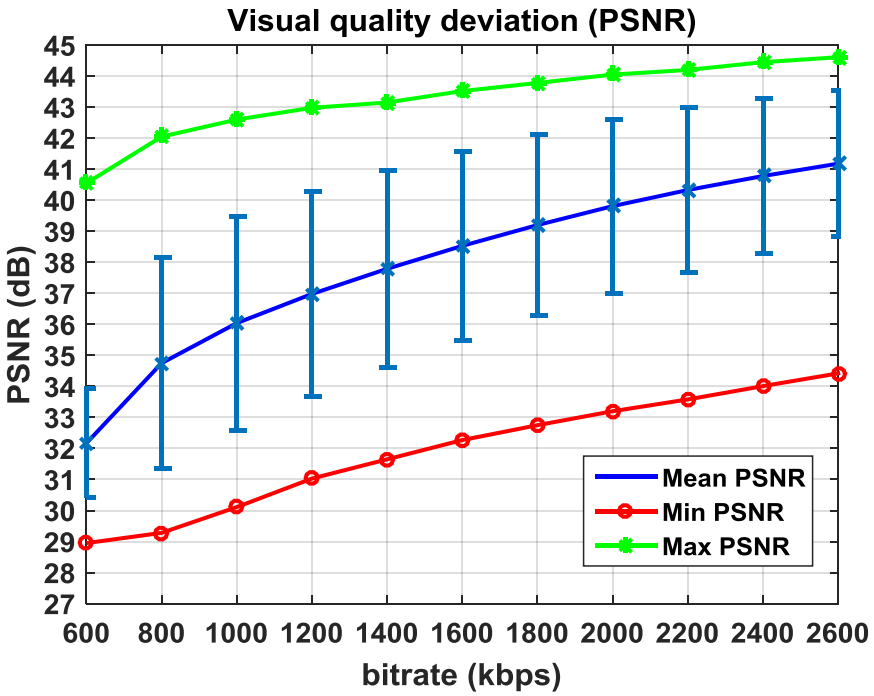


Figure 5.19: Visual quality as a function of the video bitrate. The blue line represents the mean PSNR, and the bars represent the standard deviation of the PSNR. The red line represents the minimum PSNR value encountered over the entire teleoperation session. The green line represents the maximum PSNR value encountered over the entire teleoperation session.

conclude that the multiplexed audio and video data does not cause a large increase in the packet rate. In the last 4 columns of Table 5.3, we present the visual quality of the teleoperation scenes (1280×720 @ 25 fps) in terms of mean, standard deviation, minimum and maximum PSNR. We observe that the system adapts the video bitrate, and consequently, the quality of the video stream is improved. However, it is important to note that the quality of the video is highly dependent on the motion in the video due to the camera being mounted on the robot arm (eye on the hand). When the robot is not moving, the PSNR is highest, as shown in the last column. In contrast, the PSNR is smallest if the robot experiences large motions. Furthermore, we perform an offline analysis of the visual quality of a raw recorded telemanipulation video and consider a wide range of video bitrate settings from 600 to 2600 *kbps*, which are possible bitrate settings for transmission rates ranging from 800 to 3000 *kbps*.

Fig. 5.19 illustrates the rate-distortion performance of the video encoding block [GCE+15] for the peg-in-hole task video. The blue line shows the average PSNR value of all frames in a session, and the vertical blue bars represent the standard deviation of the PSNR measurements. The red and green lines refer to the minimum and maximum PSNR values that are encountered during a peg-in-hole manipulation session. We observe that the variability of the

Table 5.4: PSNR to MOS conversion [KRW03].

PSNR (dB)	**MOS**
≥ 37	5 (Excellent)
$31 - 37$	4 (Good)
$25 - 31$	3 (Fair)
$20 - 25$	2 (Poor)
≤ 20	1 (Bad)

quality decreases as the bit budget increases. The PSNR values represented by the green line are observed when nothing moves in the teleoperation scene. The minimum PSNR values indicated by the red line are encountered if the robot and the objects interact with large motions. In [KRW03], the mapping between the PSNR values in dB and the mean opinion scores (MOS) on the ITU-T 5-point scale was derived to allocate semantic equivalents for the PSNR values on the dB scale. Table 5.4 illustrates the range of PSNR values in dB and their corresponding quality equivalence on the MOS 5-point scale. Additionally, in [Vid], the authors claimed that a picture quality of 35 dB or higher can be considered as a good quality of 4 on the MOS 5-point scale, which also confirms the mapping in Table 5.4.

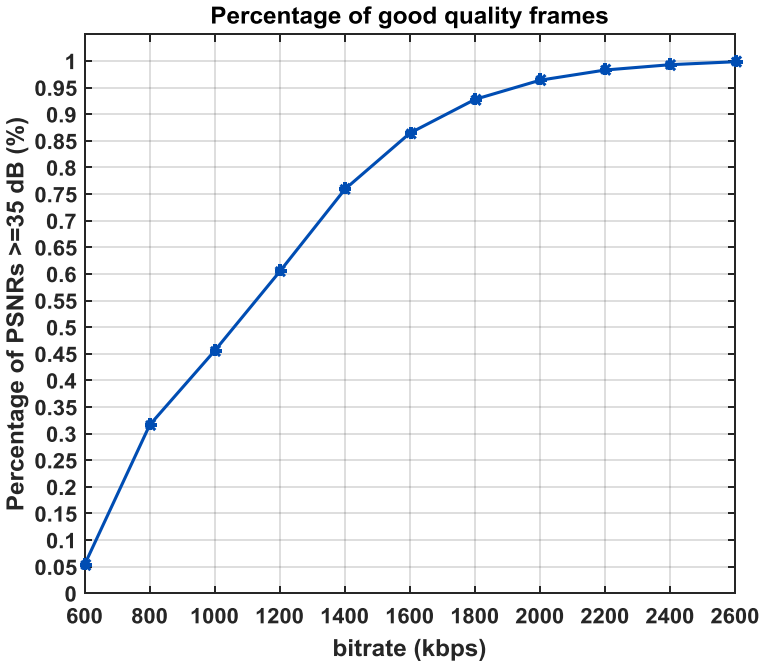


Figure 5.20: Percentage of video frames that are encoded with good quality (above 35 dB).

As a further investigation of the visual quality of the system, we measure the percentage of video frames with a PSNR higher than 35 dB in Fig. 5.20. We observe that video bitrate settings of higher than 1.8 $Mbps$ yield more than 90% good-quality pictures. After the video bitrate setting of 2.6 $Mbps$, all video frames achieve excellent quality levels. Therefore, increasing the video bitrate above 2.6 $Mbps$ does not provide any improvements in terms of visual quality, and it increases the data and packet rate unnecessarily. Hence, in this setup, we do not update the video bitrate above 2.6 $Mbps$, and the corresponding transmission rate demand of the system is approximately 3 $Mbps$ for a high-quality teleoperation. However, it is important to note that the scene complexity can be different in other applications, where a higher or lower transmission rate can be required to achieve good quality. Because our teleoperation setup is a testbed for testing a representative manipulation task for telemanipulation applications, a pre-scene analysis is needed for different scenarios to investigate the maximum required transmission rate resulting in good visual quality. Thus, the proposed procedure can be applied in a straightforward manner for any application case.

Table 5.5 presents the delay performance of the system for each modality. The time stamps of each video and audio frames and haptic samples are fed back to measure their transmission delay. The one-way 50 ms propagation delay is subtracted from the results to clearly illustrate the delay effect of the network bottleneck. The statistics of the force delay demonstrate that the multiplexing scheme successfully controls the delay constraint of the force signal. In Table 5.5, the maximum force delay values prove that the force samples are not delayed by more than the multiplexing force buffer sizes (15, 10 and 5 ms for 1, 2 and 3 $Mbps$), which is determined by Eq. 5.14. The mean and jitter values refer to the average and standard deviation of the original transmission delay of the force samples. To avoid signal distortion, a jitter-buffer is added as a play out buffer to limit the delay to the desired value at the DEMUX side. Therefore, the mean delays are shifted to the maximum delay (multiplexing buffer size), and the jitter is minimized to 0 ms. Regarding the video delay statistics, we can conclude that the linear video bitrate model determined in Section 5.3.2 performs well in controlling the video delay together with the single-frame delay constraint. The delay constraint is set to 35 ms,

Table 5.5: This table shows the delay measurements of the system for CBR links of 1, 2 and 3 $Mbps$.

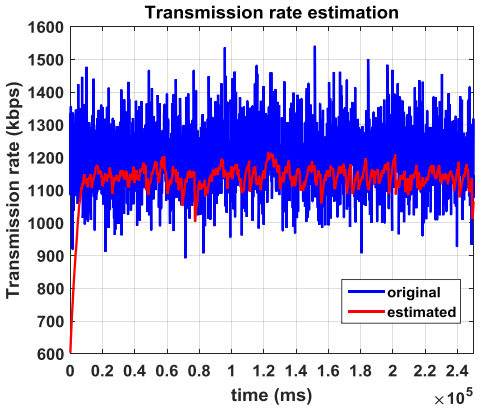
TR	**Force delay** (ms)				**Video delay** (ms)				**Audio delay** (ms)			
$(kbps)$	$mean$	$jitter$	min	max	$mean$	$jitter$	min	max	$mean$	$jitter$	min	max
1000	10.14	5.18	1	15	31.35	1.65	23.3	52.70	12.27	5.64	2.00	29.00
2000	6.47	2.88	1	10	33.05	1.13	31.40	50.50	9.89	3.57	1.60	20.20
3000	3.81	1.06	1	5	33.82	1.52	31.40	46.60	8.17	2.10	1.40	13.30

and we observe that the system converges to the constraint with very low jitter. If we check the minimum and maximum delays, outliers are sometimes found due to RC deviations for some frames. However, a low jitter value ensures that these outliers are very rare. When we consider the delay of audio frames, they can reach the DEMUX side very quickly. We observe a slight decreasing trend on the audio delay as the network capacity increases because the audio bitrate is kept constant under all transmission rate conditions as 64 $kbps$, which gives sufficient sound quality for the interaction. Hence, the audio transmission is slightly improved as the transmission rate increases.

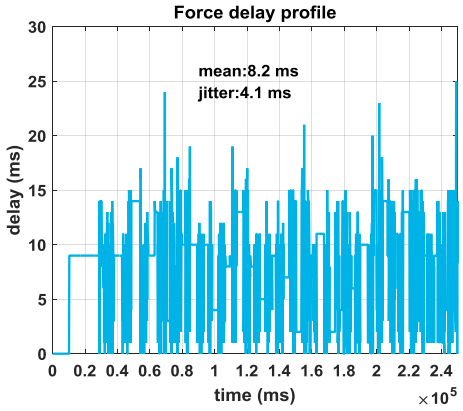
5.4.2 Experiment 2: Teleoperation with time-varying transmission capacity

In this experiment, we challenge the system with a link having a mean bitrate of 1.2 $Mbps$ and a standard deviation of 95 $kbps$, as illustrated in Fig. 5.21(a). Similarly, the RTT delay is set to a constant delay of 100 ms symmetrically. The system is run in real-time mode, and the OP actively manipulates the objects to perform the peg-in-hole task. In Fig. 5.21(a), the red line represents the estimated capacity of the link. Because the deviation of the transmission rate is very rapid, Algorithm 5 cannot predict the sudden drops and rises of the transmission rate. However, it can detect the mean capacity of the link, as observed in Fig. 5.21(a), where the red line is close to the mean bitrate of 1.2 $Mbps$. During the experiment, we measured the delay of the force, video and audio signals; the average packet rate; and the PSNR for visual quality. The resulting measurements are presented in Figures 5.21(b), 5.21(c), 5.21(d), 5.21(e) and 5.21(f), respectively. For the force delay measurements, we observe that the 15 ms delay constraint set by the MUX is violated 1.2% of the time, and the maximum force delay is measured as 25 ms throughout the entire session. When we perform the same analysis on video delay, the 35 ms delay constraint on the video frames is violated 12% of the time, and the maximum encountered video delay is 58 ms during the telemanipulation session. Regarding the audio delay, the audio frames are not affected due to the varying capacity. The mean delay and jitter of the audio frames are measured as approximately 11 and 5 ms, respectively. The average packet rate of the system is 153 $packets/second$, as observed in Fig. 5.21(e). Concerning the visual quality of the system, the average PSNR is approximately 31.34 dB (between frames 1000 - 7000, where the robot is actively manipulating), which can be accepted as a fair quality level.

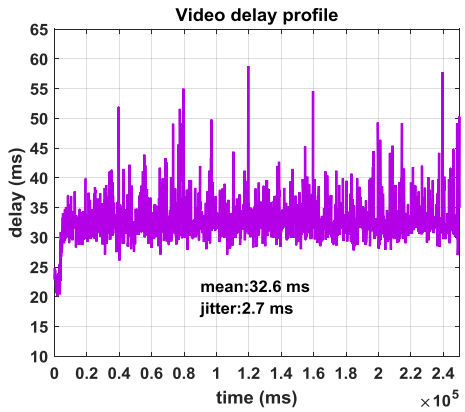
In Fig. 5.21(f), we observe very high PSNR values above 40 dB due to the static state of the robot and the objects at the beginning of the session (at approximately 30s). When the robot starts moving, the PSNR drops immediately to the fair quality level with respect to Table 5.4.



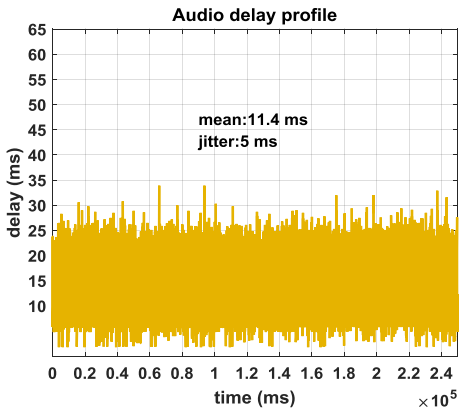
(a) Time-varying transmission rate and its estimation



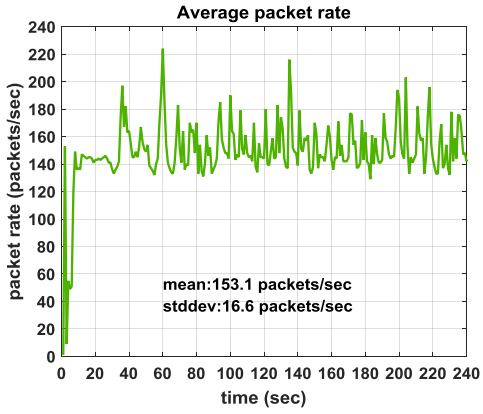
(b) Force delay



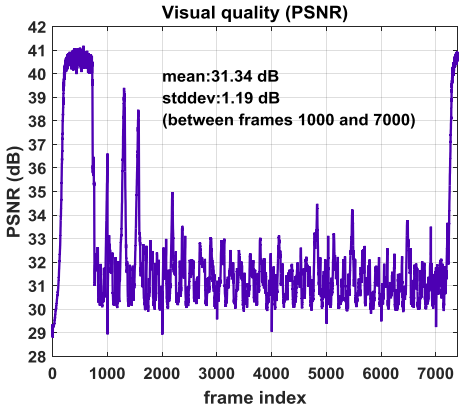
(c) Video delay



(d) Audio delay



(e) Average packet rate



(f) Frame-level PSNR measurements

Figure 5.21: Transmission rate estimation, signal delays, average packet rate and visual quality measurements for a time-varying transmission link having a mean capacity of 1.2 *Mbps*.

5.4.3 Experiment 3: Teleoperation over a CBR link shared with another session

In this section, we consider another possible scenario for teleoperation systems. In Fig. 5.22, we demonstrate an on-orbit teleoperation scenario for space missions similar to the teleoperation system between ISS (International Space Station) and DLR (German Aerospace Center) reported in [ABR+16]. Such an exploration scenario may exist for Mars exploration missions whereby astronauts in a spaceship orbiting Mars perform teleoperation on Mars. In this scenario, we present another challenge by adding two operators at the space station who are connected with two teleoperators located on the ground for a telemanipulation mission. The uplink and downlink capacities from the ground station to the space station are given as a 4 *Mbps* CBR link. On the ground, the teleoperators are connected via a wireless local area network, which provides faster connection speeds, such as 10 *Mbps*, in both directions. Let us assume that OP-1 started a teleoperation session with TOP-1. Later, OP-2 should urgently join the teleoperation event using TOP-2 to assist TOP-1 on the ground. We assume that the session belonging to TOP-1 occupies a mean bitrate of 2 *Mbps* on the channel. Thus, TOP-2 needs to predict the available capacity of the communication link. Fig. 5.23 presents the response of TOP-2 when a 4 *Mbps* CBR link is shared with TOP-1 occupying a 2 *Mbps* average bitrate using the FIFO principle. The goal of this experiment is to illustrate the system performance for unscheduled shared links. To realize this event experimentally, we

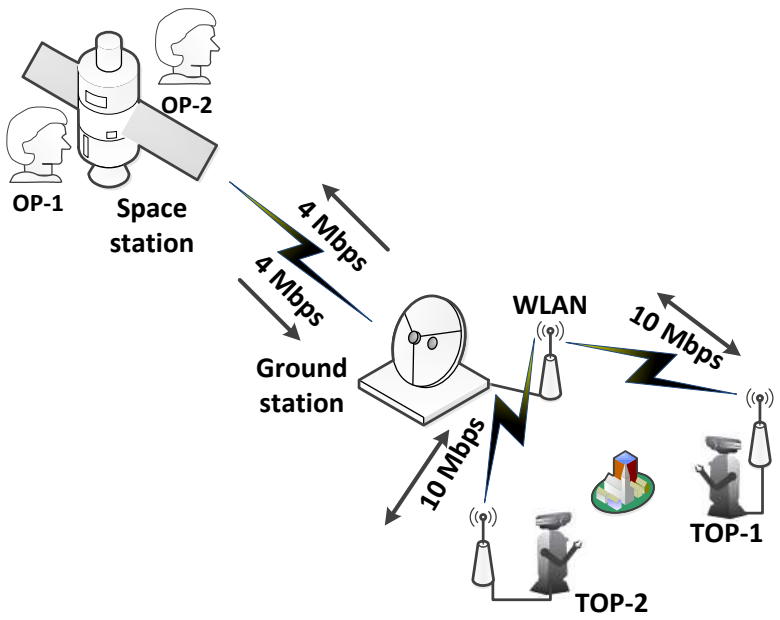


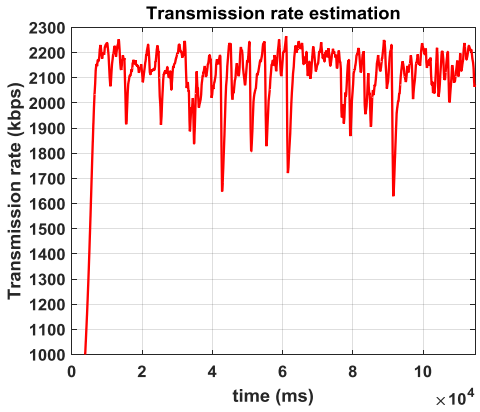
Figure 5.22: Teleoperation scenario for a space mission; the given bitrates indicate the uplink speeds.

prerecord a teleoperation session into a ".pcap" file via Wireshark [Wir] by setting the session bitrate to 2 $Mbps$. Then, we load the captured file into the network emulator to run parallel teleoperation session traffic alongside a real teleoperation stream.

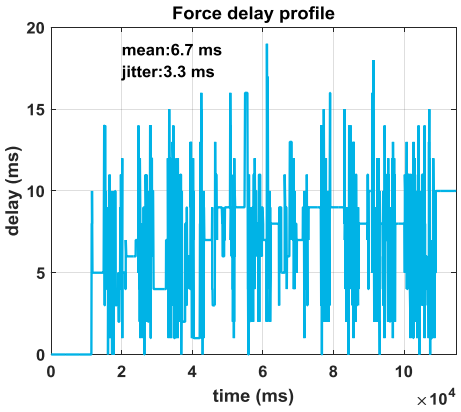
Fig. 5.23(a) shows the estimated available transmission rate for TOP-2. Both streams access the transmission medium based on the FIFO principle. When the channel is not busy with the transmission of the TOP-1 stream, the packets of TOP-2 obtain full access to the channel. On the other hand, the packets of TOP-2 have to wait if the channel is busy with the TOP-1 stream. Due to the unscheduled transmission of both streams, we observe a noisy estimation of the available channel capacity for TOP-2. Similar to the previous experiments, we also present the delay performance of the system for this scenario. Fig. 5.23(b) illustrates the force delay measurements for TOP-2. The corresponding delay constraint on the force signal is 10 ms, and 8.7% of the force samples exceed the desired delay constraint. Because the transmission medium does not apply any scheduling discipline on the streams of TOP-1 and TOP-2, the packets of TOP-2 are sometimes blocked by the packets of TOP-1. Furthermore, this causes fluctuations in the available transmission rate estimation. Therefore, the usage of a rate shaper as a scheduler is necessary to regulate the bitrate of the TOP-1 and TOP-2. The video delay profile is drawn in Fig. 5.23(c), and we see that the delay constraint of 35 ms is ensured, with minimal deviation and some rare outliers. Fig. 5.23(d) shows the audio delay profile. We observe that the audio delay can be minimized due to its low bitrate demand. In Fig. 5.23(e), the average packet rate varies, considering the rapid change in the transmission rate estimation, and it remains below 200 $packets/second$ on average. Finally, a good visual quality is achieved throughout the session, as illustrated in Fig. 5.23(f).

5.4.4 Experiment 4: Teleoperation over congested CBR links

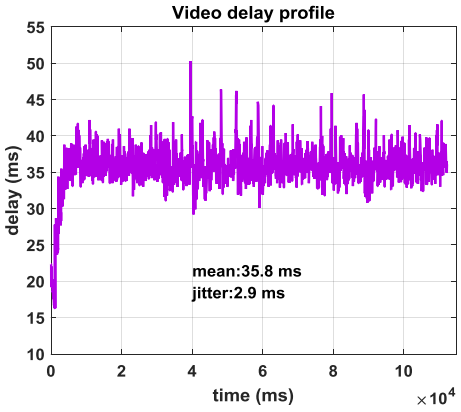
Internet service providers apply traffic shaping to regulate the incoming packet streams from different users based on the class of the customer [ITU04]. In the case whereby a high-priority customer initiates a high transmission rate-demanding application, the service provider may reduce the capacity of lower priority customers in a heavy traffic load scenario. In this experiment, we investigate the system response to sudden available transmission rate drops. As a test condition, the transmission rate of the communication link suddenly drops from 3 to 2 $Mbps$ while the OP is in contact with the object and dragging it over the surface. We test the performance of Algorithm 7, where the system detects congestion events and converges quickly back to the true communication capacity. As the RTT increases, the true transmission rate detection is delayed. Consequently, the system adaptation to the transmission rate change is also delayed. Thus, the system pushes a high data rate until it estimates the true communication capacity. During this period, the signal delay constraints are violated until the true capacity of the link is determined. In the following Figures 5.24, 5.25 and 5.26, we



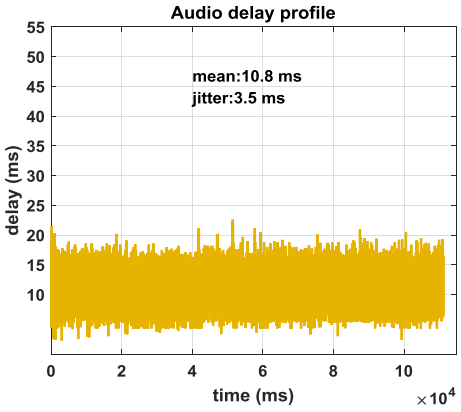
(a) Available transmission rate estimation



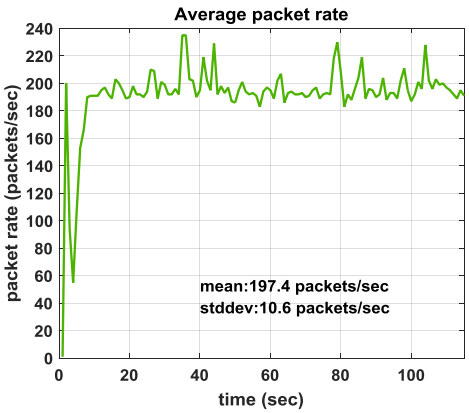
(b) Force delay



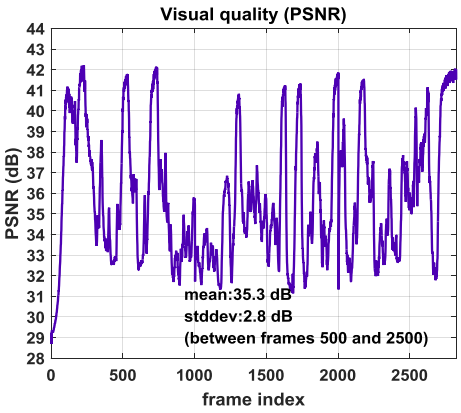
(c) Video delay



(d) Audio delay

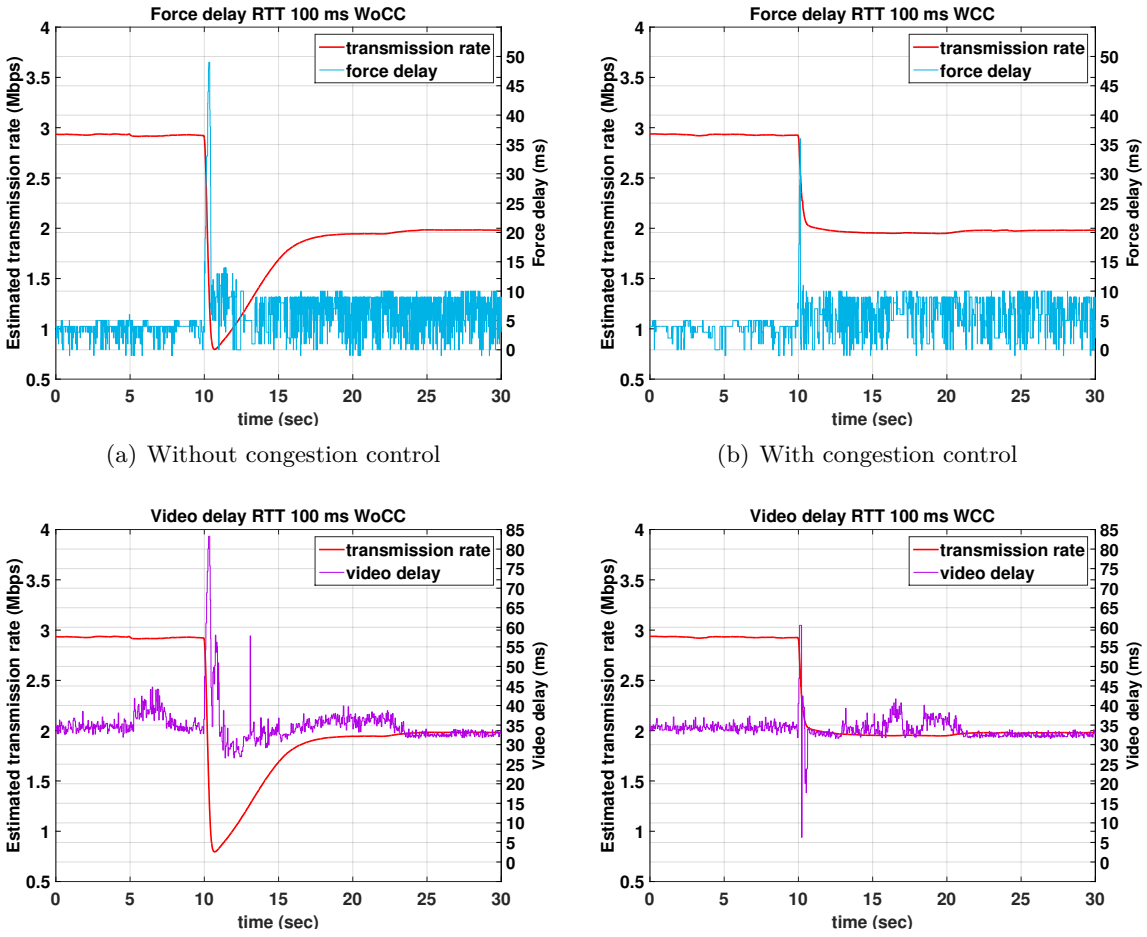


(e) Average packet rate



(f) Frame-level PSNR measurements

Figure 5.23: Teleoperation over a 4 *Mbps* CBR link shared with a prerecorded 2 *Mbps* TOP session.



(a) Without congestion control

(b) With congestion control

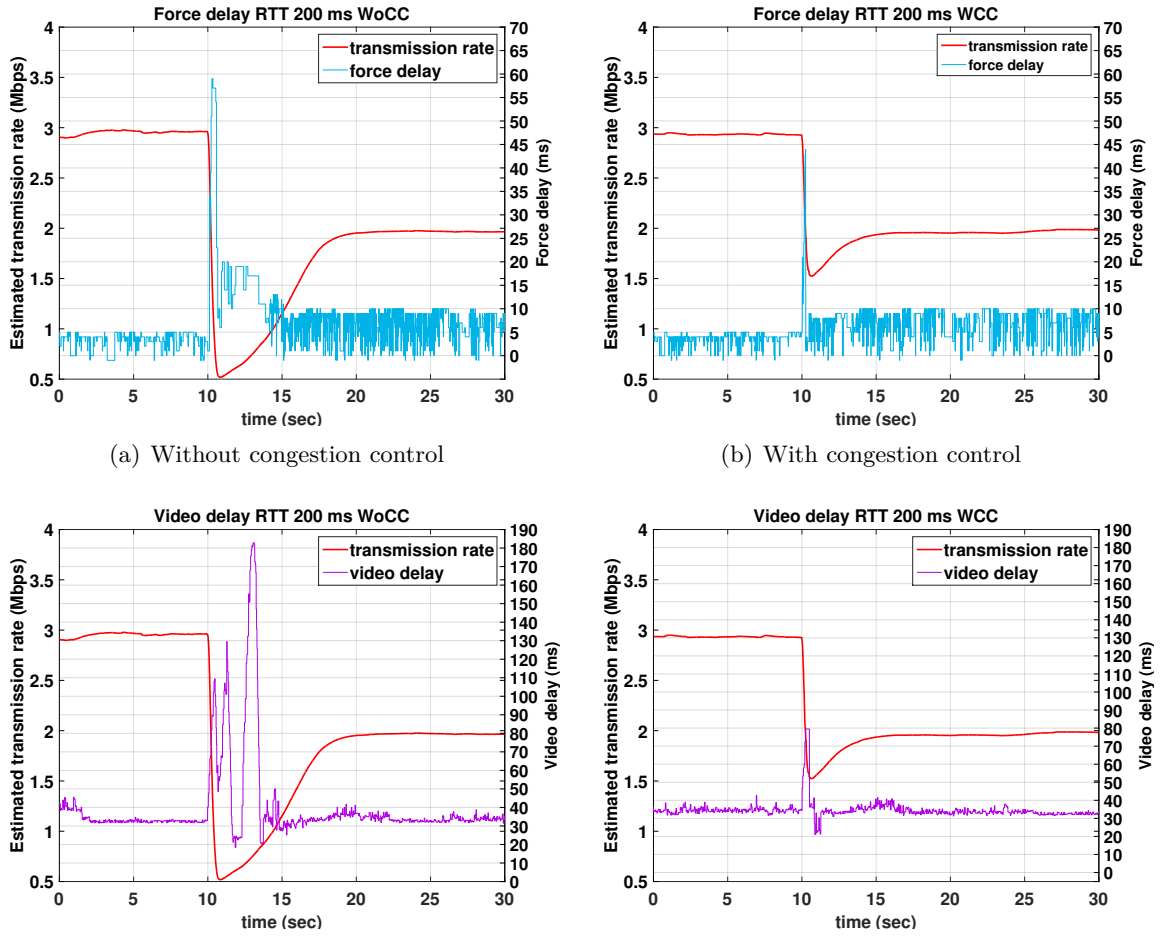
(c) Without congestion control

(d) With congestion control

Figure 5.24: The improvements of the congestion control mode when a sudden transmission rate drop occurs from 3 *Mbps* to 2 *Mbps* for a RTT of 100 *ms*.

illustrate the effect of sudden transmission rate drops on the system for RTT delays of 100, 200 and 300 *ms*. Moreover, we compare the system response when the congestion detection and control Algorithm 7 is enabled.

RTT 100 ms: Fig. 5.24 shows the estimated transmission rate results and the delay profiles of the video and force signals when the RTT delay is set to 100 *ms*. If the congestion control is disabled, we observe that the transmission rate estimation Algorithm 5 conservatively drops the throughput to below 1 *Mbps*, even though the true capacity is 2 *Mbps*. However, using Algorithm 7, the system detects the congestion event and probes the communication link capacity to converge to the true transmission rate smoothly. The red line in the figures illustrates the estimated transmission rate, and we observe that the congestion control mode helps the estimator to quickly converge to the true capacity of the link. Moreover, we draw plots for the force and video delay profiles, which are aligned with the transmission rate estimation result. We observe additional delay constraint violations if the congestion control



(a) Without congestion control

(b) With congestion control

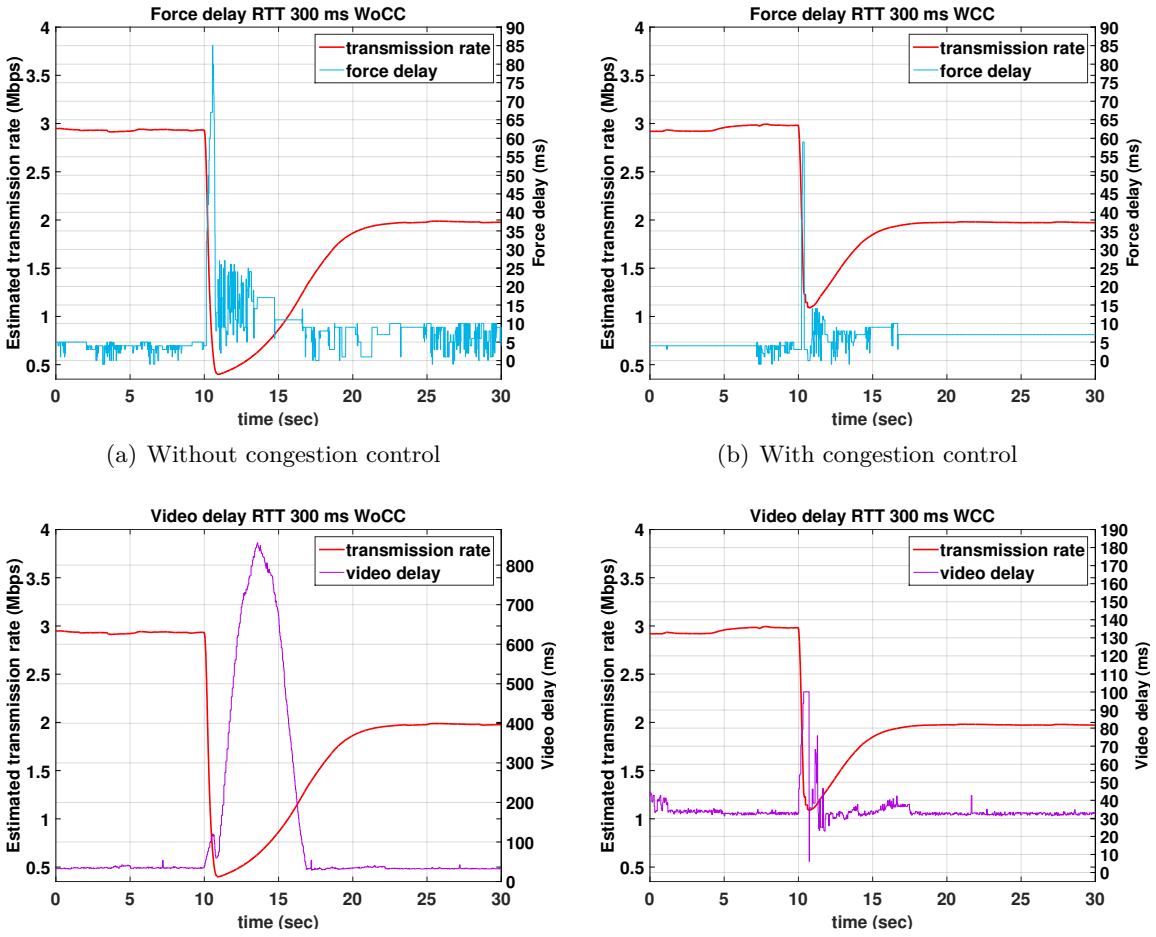
(c) Without congestion control

(d) With congestion control

Figure 5.25: The improvements of the congestion control mode when a sudden transmission rate drop occurs from 3 *Mbps* to 2 *Mbps* for a RTT of 200 *ms*.

mode is disabled. On the other hand, the peak delays do not cause critical lags for either the visual or haptic modalities.

RTT 200 ms: Fig. 5.25 illustrates the estimated transmission rate results and the delay profiles of the video and force signals when the RTT delay is set to 200 *ms*. If we compare the transmission rate estimation results in Fig. 5.25 with Fig. 5.24, we observe that the transmission rate estimation diverges very quickly from the true capacity as the RTT delay increases from 100 to 200 *ms*. On the other hand, the congestion control mode helps the estimator to converge to the true transmission rate of 2 *Mbps* by keeping the turning point above 1.5 *Mbps*. When we compare the force delay plots in Fig. 5.25(a) and Fig. 5.25(b), the congestion control mode quickly ensures that the force delay remains below the desired constraint. However, as we observe in Fig. 5.25(a), the force delay constraint of 10 *ms* is violated for a duration between 10 and 15 *seconds* if the congestion detection and control scheme is disabled. When we compare the video delay profiles in Fig. 5.25(c) and Fig.



(a) Without congestion control

(b) With congestion control

(c) Without congestion control

(d) With congestion control

Figure 5.26: For an RTT of 300 *ms*, significant improvements in the congestion control mode can be observed for a sudden transmission rate drop from 3 *Mbps* to 2 *Mbps*.

5.25(d), there is a significant improvement of the video delay when enabling the congestion detection and control scheme.

RTT 300 ms: Fig. 5.26 illustrates the estimated transmission rate results and the delay profiles of the video and force signals when the RTT delay is set to 300 *ms*. When we compare the transmission rate estimation results in Fig. 5.26 with Fig. 5.25 and Fig. 5.24, the increasing RTT delay impairs the estimation. In particular, if the congestion detection and control scheme is disabled, the estimation drops below 500 *kbps*, and it takes more than 10 *seconds* to converge to the true capacity of the link. Similar to the force delay results for the cases with an RTT delay of 100 and 200 *ms*, the congestion detection and control mode quickly converges the force delay to its desired constraint. On the other hand, the video delay diverges very quickly close to 1 *second* due to the late adaptation of the video bitrate in Fig. 5.26(c) if Algorithm 7 is disabled. However, as seen in Fig. 5.26(d), enabling Algorithm 7 helps the system to recover the video delay in a very short time without overshooting.

Table 5.6: Congestion control results when the link capacity suddenly drops from 3 to 2 $Mbps$.

RTT (ms)	**without congestion control**				**with congestion control**			
	D^{max}_{Haptic} (ms)	D^{max}_{Video} (ms)	TR_{drop} $(kbps)$	T_{conv} (ms)	D^{max}_{Haptic} (ms)	D^{max}_{Video} (ms)	TR_{drop} $(kbps)$	T_{conv} (ms)
100	49	83	1203	9921	36	60	0	1649
200	59	182	1481	10178	43	79	476	6266
300	85	845	1590	12962	59	100	903	7467

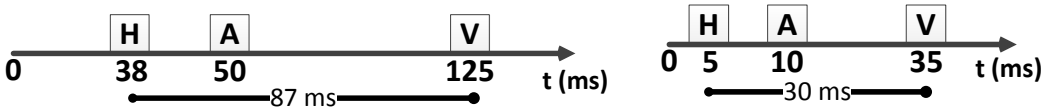
Table 5.6 summarizes the results of Figs. 5.24, 5.25 and 5.26. We compare the system responses with the congestion control Algorithm 7 disabled and enabled. We give the peak delay of the haptic and video signals, compute the transmission rate estimation drop with respect to the target bitrate of 2 $Mbps$ and measure the convergence time, which is the time difference between the time at which congestion begins and the time at which the system converges to 2 $Mbps$. From the table, we see that the congestion control reduces the system latency by controlling the video throughput of the system. On the other hand, we see that the RTT plays an important role, and it becomes challenging for the system to adapt the parameters as the RTT increases.

5.5 Discussion on the delay requirements and inter-media synchronization

In Table 5.7, we give the service provided by our teleoperation system excluding the RTT delay. From Table 5.7, we can conclude that our teleoperation system does not violate the reported latency constraints. On the other hand, the video stream can tolerate an additional 275 ms one-way propagation delay, and the audio stream can permit an additional 100 ms one-way propagation delay. However, the delay requirement on haptic signals in Table 5.7 is very tight, but this is for systems without control architectures. In our system, the delays

Table 5.7: The provided service without one-way delay and the latency requirements for haptic, video and audio streams. The bold numbers refer to the latency requirements as the maximum tolerable delay and jitter for haptic, video and audio streams given in [ECES11] and [MYY08].

Modality	**Delay(ms)**	**Jitter(ms)**
Haptic	$38 - 48 <$ **50**	$0 <$ **2**
Video	$125 <$ **400**	$3 <$ **30**
Audio	$50 <$ **150**	$5 - 10 <$ **30**



(a) with intrinsic delays

(b) without intrinsic delay (only multiplexing delay)

Figure 5.27: The inter-media synchronization.

can be tolerated up to a one-way delay of 250 *ms*. It is important to note that the latency constraints given in Table 5.7 were determined for teleconference systems (audio-video) and shared haptic virtual environments. Therefore, we need to also consider delay perception research for teleoperation systems and cross-modality effects.

Due to the bi-directional communication of position/velocity - force signals and varying environmental conditions, studying the human delay perception in haptic teleoperation systems is challenging compared to audio-visual delay perception [RSMH10]. In [Vog04], the author reported that time differences up to 45 *ms* between visual and force signals are acceptable when the subject hits a stiff wall with the haptic device. Moreover, the author reported that the perception of delay between force and visual stimuli can be larger under certain telemanipulation situations. In [RSMH10], the authors showed that the operator's movement dynamics, the local haptic device features and the penetration depth into the structures of the remote environment affect the visual-haptic delay perception. In their experiments, the delay detection thresholds were between 24 to 46 *ms*. Unlike the other publications, the authors in [SOC+13] reported that the acceptable delay between all modalities (A-V, A-H and V-H) is approximately 100 *ms*, regardless of the haptic device type. In contrast to previous studies, they performed experiments with a passive interaction, where the subject simply holds the stylus and perceives a controlled interaction. Summarizing, it is important to note that most of these previous studies did not consider the intrinsic delays in their systems, as we reported in this thesis. In Fig. 5.27, we give the maximum time difference (occurring when the force buffer size is 5 *ms*) between modalities when we consider the delays including (Fig. 5.27(a)) and excluding (Fig. 5.27(b)) the intrinsic and encoding delays. As observed in Fig. 5.27(a), the delay between video and haptic signals is approximately 87 *ms*. Under the current conditions, subjects report that they perceive the signals in synchrony and that the remote interaction is smooth for all tested RTT delay cases. On the other hand, it is important to note that the intrinsic delays and encoding time can be reduced using specialized hardware for the visual communications part of the system. If these delays are reduced, the asynchrony between the signals becomes lower than the reported thresholds in the literature given above (see Fig. 5.27(b) with the multiplexing delays only).

5.6 Chapter summary

The transmission of multimodal signals over a bandlimited communication link is a challenging problem in terms of ensuring delay requirements for each modality. In this chapter, we studied a specific transmission scenario for teleoperation systems involving audio, video and force signals and presented a novel multiplexing scheme to provide low-delay communication for audio, video and force signals. We can summarize the most important points as follows:

- An application layer flow control scheme as a multiplexer is designed for controlling the signal bitrates and system parameters to satisfy the desired delay constraints on force samples and video frames.
- An adaptive transmission rate estimator is adopted in the teleoperation system to track the instant deviations of the communication capacity.
- The latency performance of the teleoperation system is tested under various communication scenarios.
- The effect of increasing RTT delay on the transmission rate estimation and the system response is studied, and an intelligent source-skipping method is applied to compensate for sudden congestion events.

Chapter 6

Conclusion and Outlook

In this thesis, we introduced a detailed communication system setup for haptic teleoperation that can facilitate manipulation tasks under bandlimited and delayed network conditions. As a major contribution, we proposed an application layer communication protocol for teleoperation systems involving audio, video and force modalities. The proposed scheme transmits the multi-modal signals with the lowest possible latency and simultaneously efficiently utilizes the network resources. For this achievement, we employed recently developed data reduction and bitrate control techniques in the audio-visual-haptic communications context. Moreover, the teleoperation system developed in this thesis is built using a KUKA LWR arm as the TOP and a Force Dimension Omega 6 haptic device as the OP, which are commercially available robotics hardware. Therefore, it is possible for interested researchers to replicate the system and reproduce the results of this thesis for further investigations and improvements in this research field.

6.1 Accurate rate control for low-delay video communication

In a teleoperation system, the video demands high amounts of transmission rate resources and needs to be compressed efficiently to guarantee good picture quality and low frame delay. In Chapter 4, we presented an MB-level RC scheme that relies on the well-known ρ-domain rate model. We further developed an exponential model to predict the relation between the QP and ρ (the ratio of the zero coefficients after the DCT). Using the proposed exponential model, the rate allocation is accelerated in terms of computation time. Hence, it is possible to encode $720p$ video frames at 25 fps in real time. Moreover, MB-level QP smoothing is applied to avoid spatial quality deviation within a frame. On the other hand, the video encoder communicates with the multiplexing scheme to instantly adapt its encoding mode and the current bitrate setting as a response to the changing transmission rate of the communication link.

Limitations and future work

The RC scheme breaks the original algorithm flow in H.264, which increases the complexity of the codec. Additionally, it is not easy to parallelize the encoding and rate allocation into multiple threads due to the spatial MB dependency for intra prediction cases and different bitrate demands of MBs at different spatial locations. As a future extension, the second pass of the rate allocation can be merged into the first pass by predicting the texture complexity of every MB. Furthermore, the rate allocation can be divided into several slices to decrease the computation time with multiple threads, wherein each thread is responsible for encoding one slice.

6.2 Multiplexing scheme for multimodal teleoperation

In this thesis, we specifically focused on the transmission of multiple modalities of a teleoperation system running over a limited capacity link. We introduced delay-jitter effects from the low communication speed on the transmitted signals. In particular, haptic signals need to be handled with high priority due to their bidirectional exchange between the slave and master sides. Additionally, we showed that the transmission bottleneck is caused by large video packets. As a solution for the efficient transmission scheduling of audio, video and force signals, we proposed an application layer multiplexing scheme in Chapter 5. Moreover, the capacity of the communication link can change over time. Therefore, the available transmission rate of the network needs to be instantly estimated, and the system parameters, i.e., the video bitrate, multiplexing throughput and force buffer length, must be adapted according to the recent network resources. This has been achieved via the adoption of a non-linear adaptive filtering approach called TIBET [CFM04]. On the other hand, we observed that unavoidable RTT delays impair the transmission rate estimation in the case of sudden unexpected congestion or capacity drops. Additionally, the signal delays abruptly overshoot until the true capacity of the network is predicted and the system parameters are adapted to the current transmission rate condition. During this uncertain period, the operation should not be interrupted to avoid dangerous task failures. Thus, we further extended the scheme with the implementation of congestion detection and compensation procedures.

Limitations and future work

To facilitate quick prototyping and easy benchmarking, the teleoperation system in this thesis is implemented on personal computers running Ubuntu Linux OS patched with the Xenomai real-time kernel. However, this presents some drawbacks and challenges for interfacing different hardware, and the usage of a general-purpose processing unit also increases the com-

putation time for complex math calculations. Especially for high-precision transmission rate estimation, it is crucial to sample the time at a high sampling rate, and the system must process the data in a period of time that is shorter than the sampling period to keep the control loop frequency constant for real-time operation purposes. Similarly, we also observed hardware and computation delays in the video processing pipeline of the system. Additionally, the control and media processing components of the teleoperation system are distributed over several computers, which makes implementation, testing and debugging cumbersome. As a major future work, the implementation of the teleoperation system needs to be converted into specialized dedicated programmable hardware, such as an FPGA, DSP or GPU board, where control and media processing components can be placed on the same board.

The multiplexing scheme divides the available transmission rate into discrete resource buckets. If there are insufficient data available in the media buffers, the discrete transmission buckets cannot be utilized all the time. On the other hand, the irregular trigger of haptic samples makes it challenging to preempt the transmission of haptic samples. Therefore, a multiplexing buffer is applied to foresee the transmit and non-transmit states of the haptic samples. The multiplexing buffer is also a discrete queue, and its length is adapted with 5 ms discrete steps based on the the level of the available transmission rate. It is not possible to adapt the length continuously because this may lead to a loss of haptic samples due to costly queue processing operations. Regarding the available transmission rate estimation, the scheme performs best at the application layer. On the other hand, the transmission should be handled by a reliable network, and lower layers of the communication need to be optimized. For example, under the "Future Internet Engineering" project, in [Dom13], the authors proposed the Parallel Internet Data Streams Switching (PI DSS) mechanism to transmit high-speed CBR streams, which can guarantee bitrate allocation for each stream in the network. If such a service can be provided by the network, the multiplexing scheme for teleoperation can achieve the guaranteed low-delay transmission of audio, video and haptic signals.

In addition, the following scientific directions can be investigated as future work:

- Recently, multi-touch tactile sensors and actuators have been developed. The compression and transmission of tactile stimuli can be studied as a new modality extension in the framework.

- Replacing the TDPA control architecture with model-mediated teleoperation (MMT) will improve the transparency of the teleoperation system against RTT delay. In addition, MMT presents new compression and transmission research questions for the exchange of model parameters, environment structure etc. Integrating MMT into the multiplexing scheme can be studied as a future framework.

- In this thesis, we treated the network impairments as black box behaviors at the application layer and predicted the network conditions by relying on an application layer feedback mechanism. To achieve a full network-aware streaming for teleoperation systems, the transmission problem of multi-modal signals can be optimized across the communication layers. For instance, a specific communication scenario can be studied to provide a dedicated transmission solution and guaranteed QoS.

Bibliography

Publications by the author

Journal publications

[CXC+17] B. Cizmeci, X. Xu, R. Chaudhari, C. Bachhuber, N. Alt, and E. Steinbach. A multiplexing scheme for multimodal teleoperation. *ACM Transactions on Multimedia Computing and Applications (TOMM)*, 2017. [cited on page(s) 8, 10]

[XCANS14] X. Xu, B. Cizmeci, A. Al-Nuaimi, and E. Steinbach. Point cloud-based model-mediated teleoperation with dynamic and perception-based model updating. *IEEE Transactions on Instrumentation and Measurement*, 63(11):2558–2569, 2014. [cited on page(s) 10]

[XCSS16] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach. Model-mediated teleoperation: Toward stable and transparent teleoperation systems. *IEEE Access*, 4:425–449, 2016. [cited on page(s) 10, 20]

[XSCS16] X. Xu, C. Schuwerk, B. Cizmeci, and E. Steinbach. Energy prediction for teleoperation systems that combine the time domain passivity approach with perceptual deadband-based haptic data reduction. *IEEE Transactions on Haptics*, 9(4):560–573, 2016. [cited on page(s) 10]

Conference publications

[BCS12] F. Brandi, B. Cizmeci, and E. Steinbach. On the perceptual artifacts introduced by packet losses on the forward channel of haptic telemanipulation sessions. In *Springer Berlin Heidelberg, Haptics: Perception, Devices, Mobility, and Communication: International Conference, Proceedings of EuroHaptics, Part I*, pages 67–78, 2012. [cited on page(s) 10, 19]

[CCK+12] R. Chaudhari, B. Cizmeci, K. Kuchenbecker, S. Choi, and E. Steinbach. Low bitrate source-filter model based compression of vibrotactile texture signals in haptic teleoperation. In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 409–418, 2012. [cited on page(s) 10]

[CCX+14] B. Cizmeci, R. Chaudhari, X. Xu, N. Alt, and E. Steinbach. A visual-haptic multiplexing scheme for teleoperation over constant-bitrate communication links. In *Springer Berlin*

Heidelberg, Haptics: Neuroscience, Devices, Modeling, and Applications, Proceedings of EuroHaptics, volume 8619 of *Lecture Notes in Computer Science*, pages 131–138, 2014. [cited on page(s) 8, 10, 36]

[GCE+15] M. Gao, B. Cizmeci, M. Eiler, E. Steinbach, D. Zhao, and W. Gao. Macroblock level rate control for low delay H.264/AVC based video communication. In *IEEE Picture Coding Symposium (PCS)*, pages 210–215, 2015. [cited on page(s) 6, 10, 38, 42, 51, 53, 89, 127]

[PCDS14] G. Paggetti, B. Cizmeci, C. Dilliogluglil, and E. Steinbach. On the discrimination of stiffness during pressing and pinching of virtual springs. In *IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, pages 94–99, 2014. [cited on page(s) 10]

[XCS13] X. Xu, B. Cizmeci, and E. Steinbach. Point-cloud-based model-mediated teleoperation. In *IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE)*, pages 69–74, 2013. [cited on page(s) 10]

[XCSS15] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach. Haptic data reduction for time-delayed teleoperation using the time domain passivity approach. In *IEEE World Haptics Conference (WHC)*, pages 512–518, 2015. [cited on page(s) 4, 6, 8, 10, 20, 22, 23, 24, 59, 61, 124, 127]

General Publications

[ABR+16] J. Artigas, R. Balachandran, C. Riecke, M. Stelzer, B. Weber, J. H. Ryu, and A. Albu-Schaeffer. KONTUR-2: Force-feedback teleoperation from the international space station. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1166–1173, 2016. [cited on page(s) 94]

[Ace16] Acer. Acer XB270H, 144 Hz gaming display, 2016. http://www.acer.com/ac/en/GB/content/model/UM.HB0EE.A01. [cited on page(s) 12, 17]

[All16] AlliedVision. GigE Mako camera, 2016. https://www.alliedvision.com/en/support/technical-documentation/mako-g-documentation.html. [cited on page(s) 17, 85]

[App16] Apposite. Apposite netropy network emulator, 2016. http://www.apposite-tech.com/products/netropy-N61.html. [cited on page(s) 13, 85]

[Ard16] Arduino. Microcontroller, 2016. https://www.arduino.cc/. [cited on page(s) 17]

[AS89b] R. J. Anderson and M. W. Spong. Bilateral control of teleoperators with time delay. *IEEE Transactions on Automatic Control*, 34(5):494–501, 1989. [cited on page(s) 3, 20]

[BG92] D.P. Bertsekas and R.G. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, 1992. [cited on page(s) 34]

[BKS10] F. Brandi, J. Kammerl, and E. Steinbach. Error-resilient perceptual coding for networked haptic interaction. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 351–360, 2010. [cited on page(s) 19]

[Bos16] Bose. Bose Custom QC25 noise canceling headphones, 2016. https://www.bose.com. [cited on page(s) 12]

[BP95] L.S. Brakmo and L.L. Peterson. TCP Vegas: End-to-end congestion avoidance on a global internet. *IEEE Journal on Selected Areas in Communications*, 13:1465–1480, 1995. [cited on page(s) 71]

[BPB+13] J. Bohren, C. Papazov, D. Burschka, K. Krieger, S. Parusel, S. Haddadin, W. L. Shepherdson, G. D. Hager, and L. L. Whitcomb. A pilot study in vision-based augmented telemanipulation for remote assembly over high-latency networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3631–3638, 2013. [cited on page(s) 4]

[BS11] F. Brandi and E. Steinbach. Low-complexity error-resilient data reduction approach for networked haptic sessions. In *IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE)*, pages 135–140, 2011. [cited on page(s) 19]

[BS16] C. Bachhuber and E. Steinbach. A system for high precision glass-to-glass delay measurements in video communication. In *IEEE International Conference on Image Processing (ICIP)*, pages 2132–2136, 2016. [cited on page(s) 17]

[Bur96] G. C. Burdea. *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, New York, NY, USA, 1996. [cited on page(s) 19]

[Bur15] T. Burger. How fast is real-time human perception and technology, 2015. http://www.pubnub.com/blog/how-fast-is-realtime-human-perception-and-technology/. [cited on page(s) 15]

[BW80] J. T. Brebner and A. T. Welford. *Reaction Time in Personality Theory*. Academic Press, New York, NY, USA, 1980. [cited on page(s) 15]

[CAS14] R. Chaudhari, E. Altinsoy, and E. Steinbach. Haptics. In *Quality of Experience: Advanced Concepts, Applications and Methods*, pages 261–276. Springer, Heidelberg, 2014. [cited on page(s) 11, 123]

[CB94] J.E. Colgate and J.M. Brown. Factors affecting the z-width of a haptic display. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3205–3210, 1994. [cited on page(s) 4, 18]

[CBS08] N. Chopra, P. Berestesky, and M. W. Spong. Bilateral teleoperation over unreliable communication networks. *IEEE Transactions on Control Systems Technology*, 16(2):304–313, 2008. [cited on page(s) 3]

[CFM04] A. Capone, L. Fratta, and F. Martignon. Bandwidth estimation schemes for TCP over wireless networks. *IEEE Transactions on Mobile Computing*, 3(2):129–143, 2004. [cited on page(s) 9, 10, 60, 73, 74, 104, 125]

[CGM+02] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, and R. Wang. TCP Westwood: End-to-end congestion control for wired/wireless networks. *Wireless Networks*, 8(5):467–479, 2002. [cited on page(s) 71, 74]

[Che96] S. Cheshire. Latency and the quest for interactivity. White paper commissioned by Volpe Welty Asset Management, L.L.C., for the Synchronous Person-to-Person Interactive Computing Environments Meeting, San Francisco, Nov. 1996. http://www.stuartcheshire.org/papers/latencyquest.html. [cited on page(s) 15]

[CHK+09] J. Cha, Y. S. Ho, Y. Kim, J. Ryu, and I. Oakley. A framework for haptic broadcasting. *IEEE MultiMedia*, 16(3):16–27, 2009. [cited on page(s) 5, 26]

[CML+05] Z. Cen, M. Mutka, Y. Liu, A. Goradia, and N. Xi. QoS management of supermedia enhanced teleoperation via overlay networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1630–1635, 2005. [cited on page(s) 5, 25, 37]

[CMZX05] Z. Cen, M. W. Mutka, D. Zhu, and N. Xi. Improved transport service for remote sensing and control over wireless networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, pages 333–340, 2005. [cited on page(s) 5, 25]

[CSKR07] J. Cha, Y. Seo, Y. Kim, and J. Ryu. An authoring/editing framework for haptic broadcasting: Passive haptic interactions using MPEG-4 BIFS. In *IEEE Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC)*, pages 274–279, 2007. [cited on page(s) 5, 26]

[DA15] E. Dinc and O.B. Akan. More than the eye can see: Coherence time and coherence bandwidth of troposcatter links for mobile receivers. *IEEE Vehicular Technology Magazine*, 10(2):86–92, 2015. [cited on page(s) 30]

[DL09] J. Dong and N. Ling. A context-adaptive prediction scheme for parameter estimation in H.264/AVC macroblock layer rate control. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(8):1108–1117, 2009. [cited on page(s) 6]

[Dom13] J. Domzal. Flow-aware networking as an architecture for the IPv6 QoS parallel internet. In *Australasian Telecommunication Networks and Applications Conference (ATNAC)*, pages 30–35, 2013. [cited on page(s) 105]

[DvBA08] M. U. Demircin, P. van Beek, and Y. Altunbasak. Delay-constrained and R-D optimized transrating for high-definition video streaming over WLANs. *IEEE Transactions on Multimedia*, 10(6):1155–1168, 2008. [cited on page(s) 10, 79]

[ECES11] M A. Eid, J. Cha, and A. El-Saddik. Admux: An adaptive multiplexer for haptic-audio-visual data communication. *IEEE Transactions on Instrumentation and Measurement*, 60(1):21–31, 2011. [cited on page(s) 5, 6, 26, 38, 100, 127]

[EYY08] K. Endoh, K. Yoshida, and T. Yakoh. Low delay live video streaming system for interactive use. In *6th IEEE International Conference on Industrial Informatics (INDIN)*, pages 1481–1486, 2008. [cited on page(s) 42]

[Fer65] W. R. Ferrell. Remote manipulation with transmission delay. *IEEE Transactions on Human Factors in Electronics*, HFE-6(1):24–32, Sept. 1965. [cited on page(s) 2, 3]

[FI05] M. Fujimoto and Y. Ishibashi. Packetization interval of haptic media in networked virtual environments. In *Proceedings of 4th ACM SIGCOMM Workshop on Network and System Support for Games*, NetGames, pages 1–6. ACM, 2005. [cited on page(s) 18]

[For01] ForceDimension. Omega 6, 6-DoF haptic device, 2001. http://www.forcedimension.com/. [cited on page(s) 12, 16, 85]

[FPB10] D. Feth, A. Peer, and M. Buss. Incorporating human haptic interaction models into teleoperation systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4257–4262, 2010. [cited on page(s) 3]

[GOU+12] M. Goeller, J. Oberlaender, K. Uhl, A. Roennau, and R. Dillmann. Modular robots for on-orbit satellite servicing. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2018–2023, 2012. [cited on page(s) 3, 30]

[HB07] S. Hirche and M. Buss. Transparent data reduction in networked telepresence and teleaction systems. Part II: Time-delayed communication. *Presence: Teleoperators and Virtual Environments*, 16(5):532–542, 2007. [cited on page(s) 4, 20]

[HBS99] C. H. Ho, C. Basdogan, and M. A. Srinivasan. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence: Teleoperators and Virtual Environments*, 8(5):477–491, 1999. [cited on page(s) 3]

[HHC+08] P. Hinterseer, S. Hirche, S. Chaudhuri, E. Steinbach, and M. Buss. Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE Transactions on Signal Processing*, 56(2):588–597, 2008. [cited on page(s) 4, 6, 8, 19, 20, 38, 88]

[HHSB05] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss. Network traffic reduction in haptic telepresence systems by deadband control. In *16th IFAC World Congress*, pages p.77–82, 2005. [cited on page(s) 4, 18]

[HHSB07] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss. Transparent data reduction in networked telepresence and teleaction systems. Part I: Communication without time delay. *Presence: Teleoperators and Virtual Environments*, 16(5):523–531, 2007. [cited on page(s) 4]

[HKM01] Z. He, Y. K. Kim, and S. K. Mitra. Low-delay rate control for DCT video coding via ρ-domain source modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(8):928–940, 2001. [cited on page(s) 6]

[HM02a] Z. He and S. K. Mitra. Optimum bit allocation and accurate rate control for video coding via ρ-domain source modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(10):840–849, 2002. [cited on page(s) 6]

[HM02b] Z. He and S.K. Mitra. A linear source model and a unified rate control algorithm for DCT video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(11):970–982, 2002. [cited on page(s) 6, 8, 41, 42, 44, 55]

[HR01] B. Hannaford and J. Ryu. Time domain passivity control of haptic interfaces. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1863–1869, 2001. [cited on page(s) 3]

[HSHB05] P. Hinterseer, E. Steinbach, S. Hirche, and M. Buss. A novel, psychophysically motivated transmission approach for haptic data streams in telepresence and teleaction systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1097–1100, 2005. [cited on page(s) 4, 18, 19]

[Hum07] Humanbenchmark. Reaction time test, 2007. http://www.humanbenchmark.com/tests/reactiontime. [cited on page(s) 15]

[HW08] Z. He and D. O. Wu. Linear rate control and optimum statistical multiplexing for H.264 video broadcast. *IEEE Transactions on Multimedia*, 10(7):1237–1249, 2008. [cited on page(s) 6, 51, 53, 55, 127]

[HZS01] K. Hashtrudi-Zaad and S. E. Salcudean. Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators. *The International Journal of Robotics Research*, 20(6):419–445, 2001. [cited on page(s) 13]

[ITN11] E. Isomura, S. Tasaka, and T. Nunome. QoE enhancement in audiovisual and haptic interactive IP communications by media adaptive intra-stream synchronization. In *TENCON IEEE Region 10 Conference*, pages 1085–1089, 2011. [cited on page(s) 5, 6, 26, 38]

[ITU93] ITU-T. ITU-T JPEG Standard, Digital compression and coding of continuous-tone still images, 1993. https://www.w3.org/Graphics/JPEG/itu-t81.pdf. [cited on page(s) 27]

[ITU04] ITU-T. ITU-T Recommendation I.371:, Traffic control and congestion control in B-ISDN, 2004. http://www.itu.int/rec/T-REC-I.371-200403-I/en. [cited on page(s) 95]

[ITU05] ITU-T. ITU-T H.264, Advanced video coding for generic audiovisual services, 2005. https://www.itu.int/rec/T-REC-H.264. [cited on page(s) 6, 41]

[Ive15] V.B. Iversen. *Teletraffic Engineering and Network Planning*. DTU Fotonik, Lyngby, Denmark, 2015. [cited on page(s) 33]

[JL06] M. Jiang and N. Ling. Low-delay rate control for real-time H.264/AVC video coding. *IEEE Transactions on Multimedia*, 8(3):467–477, 2006. [cited on page(s) 6, 48, 49]

[Joi] Joint Video Team. ITU-T H.264, JVT reference software encoder. http://iphome.hhi.de/suehring/tml/. [cited on page(s) 6]

[JR383] JR3. 6-DoF force-torque sensor, 1983. http://www.jr3.com/. [cited on page(s) 13, 16, 85]

[KCL+03] S. Kang, C. Cho, J. Lee, D. Ryu, C. Park, K. Shin, and M. Kim. ROBHAZ-DT2: Design and integration of passive double tracked mobile manipulator system for explosive ordnance disposal. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Proceedings*, volume 3, pages 2624–2629, 2003. [cited on page(s) 3]

[KKHB06] M. Kuschel, P. Kremer, S. Hirche, and M. Buss. Lossy data reduction methods for haptic telepresence systems. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, 2006. [cited on page(s) 4, 18]

[KNT15] S. Kaede, T. Nunome, and S. Tasaka. QoE enhancement of audiovisual and haptic interactive IP communications by user-assistance. In *IEEE 18th International Conference on Computational Science and Engineering (CSE)*, pages 35–42, 2015. [cited on page(s) 5, 6, 26]

[KRW03] J. Klaue, B. Rathke, and A. Wolisz. Evalvid - a framework for video transmission and quality evaluation. In *In Proceedings of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 255–272, 2003. [cited on page(s) 90, 127]

[KSK07] D. K. Kwon, M. Y. Shen, and C. C. J. Kuo. Rate control for H.264 video with enhanced rate and distortion models. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(5):517–529, 2007. [cited on page(s) 6]

[KUK] KUKA. KUKA LWR, lightweight robot arm. http://www.kuka.com/. [cited on page(s) 13, 85]

[KVN+10] J. Kammerl, I. Vittorias, V. Nitsch, E. Steinbach, and S. Hirche. Perception-based data reduction for haptic force-feedback signals using velocity-adaptive deadbands. *Presence: Teleoperators and Virtual Environments*, 19(5):450–462, 2010. [cited on page(s) 19]

[Law93] D.A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637, 1993. [cited on page(s) 20]

[LCZ97] H. Lee, T. Chiang, and Y. Zhang. Scalable rate control for very low bit rate (VLBR) video. In *International Conference on Image Processing, 1997. Proceedings*, volume 2, pages 768–771, 1997. [cited on page(s) 43]

[LCZ00] H. Lee, T. Chiang, and Y. Zhang. Scalable rate control for MPEG-4 video. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(6):878–894, 2000. [cited on page(s) 43]

[LGLC10] M. Liu, Y. Guo, H. Li, and C. W. Chen. Low-complexity rate control based on ρ-domain model for scalable video coding. In *IEEE International Conference on Image Processing (ICIP)*, pages 1277–1280, 2010. [cited on page(s) 6]

[LGP+06] Z.G. Li, W. Gao, F. Pan, S.W. Ma, K.P. Lim, G.N. Feng, X. Lin, S. Rahardja, H.Q. Lu, and Y. Lu. Adaptive rate control for H.264. *Journal of Visual Communication and Image Representation*, 17(2):376–406, 2006. Introduction: Special Issue on emerging H.264/AVC video coding standard. [cited on page(s) 6]

[Lof94] R. B. Loftin. Virtual environments for aerospace training. In *WESCON/94. Idea/Microelectronics. Conference Record*, pages 384–387, 1994. [cited on page(s) 3]

[MGWL03] S. Ma, W. Gao, F. Wu, and Y. Lu. Rate control for JVT video coding scheme with HRD considerations. In *Proceedings International Conference on Image Processing*, volume 3, pages III–793, 2003. [cited on page(s) 6]

[Mil68] R.B. Miller. Response time in man-computer conversational transactions. In *ACM Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 267–277, 1968. [cited on page(s) 14]

[MN08] P. Mitra and G. Niemeyer. Model-mediated telemanipulation. *International Journal of Robotics Research*, 27(2):253–262, 2008. [cited on page(s) 3]

[MV06] L. Merritt and R. Vanam. x264: A high performance H.264/AVC encoder. 2006. http://www.videolan.org/developers/x264.html. [cited on page(s) 6, 8, 41, 51]

[MV07] L. Merritt and R. Vanam. Improved rate control and motion estimation for H.264 encoder. In *IEEE International Conference on Image Processing (ICIP)*, volume 5, pages 309–312, 2007. [cited on page(s) 8, 54]

[MYY08] A. Marshall, K.M. Yap, and W. Yu. Providing QoS for networked peers in distributed haptic virtual environments. *Hindawi, Journal of Advances in Multimedia*, 2008. [cited on page(s) 100, 127]

[NS91] G. Niemeyer and J. J. E. Slotine. Stable adaptive teleoperation. *IEEE Journal of Oceanic Engineering*, 16(1):152–162, 1991. [cited on page(s) 3, 4]

[OEIS07] H. A. Osman, M. Eid, R. Iglesias, and A. E. Saddik. ALPHAN: Application layer protocol for haptic networking. In *IEEE International Workshop on Haptic, Audio and Visual Environments and Games (HAVE)*, pages 96–101, 2007. [cited on page(s) 5, 6]

[OTT+95] M. Ouhyoung, W. N. Tsai, M. C. Tsai, J. R. Wu, C. H. Huang, and T. J. Yang. A low-cost force feedback joystick and its use in PC video games. *IEEE Transactions on Consumer Electronics*, 41(3):787–794, 1995. [cited on page(s) 3]

[PHP+07] J. Pyke, M. Hart, V. Popov, R.D. Harris, and S. McGrath. A tele-ultrasound system for real-time medical imaging in resource-limited settings. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3094–3097, 2007. [cited on page(s) 30]

[PNK76] M.I. Posner, M.J. Nissen, and R.M. Klein. Visual dominance: An information-processing account of its origins and significance. *Psychological Review*, 83(2):157–171, 1976. [cited on page(s) 15]

[Por04] PortAudio. Portable cross platform audio I/O, 2004. http://portaudio.com/. [cited on page(s) 16]

[PPB10a] C. Passenberg, A. Peer, and M. Buss. Model-mediated teleoperation for multi-operator multi-robot systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4263–4268, 2010. [cited on page(s) 4]

[PPB10b] C. Passenberg, A. Peer, and M. Buss. A survey of environment-, operator-, and task-adapted controllers for teleoperation systems. *Mechatronics*, 20(7):787–801, 2010. [cited on page(s) 4]

[PWHM14] M.C. Potter, B. Wyble, C.E. Hagmann, and E.S. McCourt. Detecting meaning in RSVP at 13 ms per picture. *Springer US, Attention, Perception, and Psychophysics*, 76(2):270–279, 2014. [cited on page(s) 15]

[PWZ05] L. Ping, L. Wenjuan, and S. Zengqi. Transport layer protocol reconfiguration for network-based robot control system. In *IEEE Proceedings of Networking, Sensing and Control*, pages 1049–1053, 2005. [cited on page(s) 4, 25, 37]

[RAP10] J. Ryu, J. Artigas, and C. Preusche. A passive bilateral control scheme for a teleoperator with time-varying communication delay. *Mechatronics*, 20(7):812–823, 2010. [cited on page(s) 3, 4, 20, 21, 23]

[RCL99] J. Ribas-Corbera and S. Lei. Rate control in DCT video coding for low-delay communications. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(1):172–185, 1999. [cited on page(s) 42]

[RKH04] J. Ryu, D. Kwon, and B. Hannaford. Stability guaranteed control: Time domain passivity approach. *IEEE Transactions on Control Systems Technology*, 12(6):860–868, 2004. [cited on page(s) 3]

[RSMH10] M. Rank, Z. Shi, H. Müller, and S. Hirche. Perception of delay in haptic telepresence systems. *Presence: Teleoperators and Virtual Environments*, 19(5):389–399, 2010. [cited on page(s) 101]

[She93] T. B. Sheridan. Space teleoperation through time delay: Review and prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, 1993. [cited on page(s) 3]

[SHE+12] E. Steinbach, S. Hirche, M. Ernst, F. Brandi, R. Chaudhari, J. Kammerl, and I. Vittorias. Haptic communications. *Proceedings of the IEEE*, 100(4):937–956, 2012. [cited on page(s) 4]

[SHK+11] E. Steinbach, S. Hirche, J. Kammerl, I. Vittorias, and R. Chaudhari. Haptic data compression and communication. *IEEE Signal Processing Magazine*, 28(1):87–96, 2011. [cited on page(s) 19, 123]

[SOC+13] J.M. Silva, M. Orozco, J. Cha, A. El-Saddik, and E.M. Petriu. Human perception of haptic-to-video and haptic-to-audio skew in multimedia applications. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(2):9, 2013. [cited on page(s) 101]

[SSZ98] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. *ACM SIGCOMM Computer Communication Review*, 28(4):118–130, 1998. [cited on page(s) 71, 75]

[Sul15] Y. Sulema. Haptic interaction in educational applications. In *International Conference on Interactive Mobile Communication Technologies and Learning (IMCL)*, pages 312–314, 2015. [cited on page(s) 3]

[Tav08] M. Tavakoli. *Haptics For Teleoperated Surgical Robotic Systems*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1st edition, 2008. [cited on page(s) 3]

[TC97] F. Tendick and M. C. Cavusoglu. Human-machine interfaces for minimally invasive surgery. In *Engineering in Medicine and Biology Society. Proceedings of the 19th Annual International Conference of the IEEE*, volume 6, pages 2771–2776, 1997. [cited on page(s) 3]

[UY04] Y. Uchimura and T. Yakoh. Bilateral robot system on the real-time network structure. *IEEE Transactions on Industrial Electronics*, 51(5):940–946, 2004. [cited on page(s) 25]

[Vid] Video Clarity. Understanding MOS, JND and PSNR. White paper. http://videoclarity.com/wpunderstandingjnddmospsnr/. [cited on page(s) 90]

[VKHS09] I. Vittorias, J. Kammerl, S. Hirche, and E. Steinbach. Perceptual coding of haptic data in time-delayed teleoperation. In *3rd Joint EuroHaptics Conference, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics*, pages 208–213, 2009. [cited on page(s) 20, 23, 24]

[Vog04] I. Vogels. Detection of temporal delays in visual-haptic interfaces. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(1):118–134, 2004. [cited on page(s) 101]

[VTMM10] J.M. Valin, T.B. Terriberry, C. Montgomery, and G. Maxwell. A high-quality speech and audio codec with less than 10-ms delay. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(1):58–67, 2010. [cited on page(s) 6, 16, 38, 59]

[Wal05] J. Walraevens. *Discrete-Time Queueing Models with Priorities*. PhD thesis, 2005. [cited on page(s) 31]

[WBBN12] B. Willaert, J. Bohg, H. Van Brussel, and G. Niemeyer. Towards multi-DoF model mediated teleoperation: Using vision to augment feedback. In *IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE)*, pages 25–31, 2012. [cited on page(s) 3]

[Web51] E. Weber. *Die Lehre vom Tastsinn und Gemeingefühl, auf Versuche gegruendet*. Vieweg, Braunschweig, Germany, 1851. [cited on page(s) 4, 18]

[Wir] Wireshark. Wireshark network protocol analyzer. https://www.wireshark.org/. [cited on page(s) 95]

[YTY13] D. Yashiro, D. Tian, and T. Yakoh. End-to-end flow control for visual-haptic communication in the presence of bandwidth change. *Electronics and Communications in Japan*, 96(11):26–34, 2013. [cited on page(s) 5, 6, 26, 38]

[YYYK14] S. Yamamoto, D. Yashiro, K. Yubai, and S. Komada. Rate control based on queuing state observer for visual-haptic communication. In *IEEE 13th International Workshop on Advanced Motion Control (AMC)*, pages 569–574, 2014. [cited on page(s) 5, 6, 26, 38]

[ZS11] F. Zhang and E. Steinbach. Improved ρ-domain rate control with accurate header size estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 813–816, 2011. [cited on page(s) 6, 41, 42]

List of Abbreviations

Abbreviation	Description	Definition
OP	Operator	page 2
TOP	Teleoperator	page 2
DoF	Degree-of-Freedom	page 12
QoS	Quality of service	page 13
UDP	Universal datagram protocol	page 13
IPv4	Internet protocol version 4	page 13
LWR	Light weight robot	page 13
RT	Real time	page 14
HCI	Human computer interaction	page 14
CELT	Constrained-energy lapped tansform	page 16
CBR	Constant bitrate	page 51
LED	Light-emitting diode	page 17
GigE	Gigabit ethernet	page 17
HD	High definition	page 17
JND	Just noticeable differences	page 18
ZOH	Zero order hold	page 19
DB	Deadband	page 19
TDPA	Time-domain passivity approach	page 20
PC	Passivity control	page 21

Abbreviation	Description	Definition
PO	Passivity observer	page 23
WV	Wave variables	page 24
FCFS	First come first serve	page 30
ATM	Asynchronous transfer mode	page 31
D/G/1	deterministic arrival/general service time/single server queue model	page 33
D/D/1	deterministic arrival/deterministic service time/single server queue model	page 33
TCP	Transmission control protocol	page 25
QoE	Quality of experience	page 26
RTT	Round trip time	page 38
RC	Rate control	page 41
AVC	Advanced video codec	page 41
RD	Rate distortion	page 41
QP	Quantization parameter	page 42
MB	Macroblock	page 43
DCT	Discrete cosine transform	page 44
MAD	Mean absolute difference	page 43
CIF	Common intermediate format	page 45
RDO	Rate distortion optimization	page 46
PSNR	Peak signal to noise ration	page 51
FIFO	First in first out	page 59
MUX	Multiplexer	page 59
DEMUX	Demultiplexer	page 59
MAC	Media access control	page 63
MTU	Maximum transmission unit	page 64
TIBET	Time-intervals-based bandwidth estimation technique	page 74

Abbreviation	Description	Definition
IIR	Infinite impulse response	page 74

List of Figures

List of Tables