Virtual Queues for QoS Compliance of Haptic Data Streams in Teleoperation

Osel Lhamo*, Giang T. Nguyen†‡ and Frank H. P. Fitzek*‡
* Deutsche Telekom Chair of Communication Networks, TU Dresden, Germany
† Haptic Communication Systems, TU Dresden, Germany
‡ Centre for Tactile Internet with Human-in-the-Loop (CeTI)
E-mails:osel.lhamo@tu-dresden.de

***Abstract*—Tactile Internet aims at allowing perceived real-time interactions between humans and machines. This requires satisfying a stringent latency requirement of haptic data streams whose data rates vary drastically as the results of perceptual codecs. This introduces a complex problem for the underlying network infrastructure to fulfill the pre-defined level of Quality of Service (QoS). However, novel networking hardware with data plane programming capability allows processing packets differently and opens up a new opportunity. For example, a dynamic and network-aware resource management strategy can help satisfy the QoS requirements of different priority flows without wasting precious bandwidth. This paper introduces virtual queues for service differentiation between different types of traffic streams, leveraging protocol independent switch architecture (PISA). We propose coordinating the management of all the queues and dynamically adapting their sizes to minimize packet loss and delay due to network congestion and ensure QoS compliance.**

***Index Terms*—Data plane programming, software defined network, P4, virtual queue, haptic data**

I. Introduction

Introducing haptic information to traditional audio-video applications can enhance human collaborations and allows for better performance with a stronger sense of togetherness among the collaborators. This growing trend of immersive applications and tactile internet has made it imperative to have systems that can support the extremely delay-sensitive haptic data along with other media streams.

Establishing dedicated networks to only serve haptic teleoperation could be practically infeasible [1]. Thus, it is implemented over the prevailing Intenet. However, the Internet has to carry several unpredictable and time-varying traffic flows from different sources to different destinations. This makes it susceptible to network overload and congestion that causes significant delays, jitter, and packet losses, resulting in severe Quality of Service (QoS) breaches. Having reliable communication that maintains extremely stringent QoS requirements of haptic data and other real-time applications over such a network is not easy. This paper proposes a solution to maintain QoS for different traffic flows even under network issues such as congestion.

Simply reserving bandwidth of peak haptic throughput can pose a problem considering the resource-constrained nature of some networks, such as quickly established ad-hoc networks for emergency operations in rural regions. Instead of a static bandwidth reservation scheme, we propose an algorithm to manage bandwidth consumption dynamically. We exploit the data plane programmability of protocol-independent switch architecture (PISA) to differentiate the various types of traffic. The idea is to separate virtual queues (or slices) and customize each slice to enable successful haptic teleoperation that satisfies the QoS of all the involved media streams. Fig. 1 depicts a high-level overview of our system and the different data flows that are present. Additionally, we set up an in-band network telemetry-based congestion control scheme to pace the TCP flows. For realizing our algorithm, we use P4, a packet processing language that manipulates the data plane of a programmable forwarding target such as a hardware or software switch, network interface card, router, or network appliance [2].

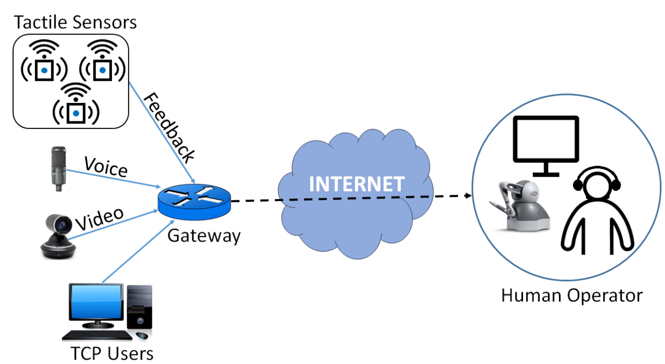

Fig. 1. Preliminary high-level architecture of the proposed system

The remainder of the paper is organized as follows. Section II briefly investigates the nature of haptic data and its compression model. Section III discusses the related work to our research. Section IV covers our proposed system architecture and associated technical concepts. It also outlines the evaluation method we use. Finally, section V briefly summarizes our paper and notes possible future work.

II. Haptic Data Compression

To guarantee the tracking performance and stability of the local control loop, haptic signals are sampled at a high rate (like 1KHz). Furthermore, in view of the considerable number of sensors involved in a haptic application, simply sending packets at such high rates over the network is not very bandwidth efficient. Consequently, many compression and reduction models have been designed for haptic applications, amongst which adaptive sampling is a commonly acknowledged model. In adaptive sampling, a haptic symbol can only be transmitted when the percent change in its value compared to a specific reference crosses a set threshold, thus, it can help cut the time-average data rate by up to 90% without impairing human perception [3]. In light of this, works like [4] [5] suggest reserving a time-average data rate as bandwidth for sending haptic samples to ensure QoS compliance. However, as shown in [1], allocating a fixed bandwidth of a low time-average rate cannot sufficiently account for the instantaneous rate, which can occasionally reach peak values much higher than the reserved bandwidth causing QoS violation. Furthermore, the data rate compression from such a sampling method is not reliant on network conditions but solely on how fast the haptic signal changes- causing a high data rate for rapidly-varying signals and conserving data rate for slow-changing symbols. We suggest devising a cognizant network to alleviate the congestion problem since the involved network elements (e.g., router, switch, etc.) can use information (metadata) observed when processing packets to make optimum decisions for avoiding or handling traffic congestion.

III. Related Work

Under this section, we first discuss some papers addressing the prevalent network congestion problem since QoS violations for different applications (particularly haptic) occur due to unpredictable but highly possible traffic overload. Then we refer to some papers from which we derived inspiration for our proposed solution.

A. Congestion Control

Over the years, many investigations focus on QoS compliance via resource management. These works include proposing algorithms to address unforeseeable network issues like congestion. In particular, due to the ubiquity of TCP traffic, many researchers focus on improving its congestion control mechanism. Morita et al. [6] proposed an adaptive queue management mechanism to tackle microbursts in data center networks using different ECN thresholds for long and short flow queues. The authors in [7] also apply ECN to dynamically adjust the threshold and prioritize small flows over large ones to ensure flow completion time, with the highest priority being given to acknowledgment signals for rapid feedback. Li et al. [8] presented High Precision Congestion Control (HPCC), a congestion control solution that leverages in-network telemetry (INT) to find precise link load information and accurately regulate traffic. HPCC can swiftly converge to use free bandwidth while avoiding congestion and maintains near-zero in-network queues by tackling difficulties such as delayed INT packets during congestion and overreaction to INT data.

Like aforementioned papers, many congestion control research works consider only data center networks that occur in supervised conditions. The real challenge is complying with strict QoS requirements of the haptic application over existing shared networks (like the Internet). As discussed in Section II, the haptic community focuses on data reduction and compression to mitigate network congestion over Internet. But these compression models lack network awareness so we suggest bringing intelligence within the network to enhance them. This is possible with the advent of Software Defined Networks (SDN) that separate the data plane and control plane to facilitate a dynamic, flexible, and cost-effective system structure. While SDN brings flexibility to the control plane, its insufficient in providing a more programmable data plane where the actual packet forwarding takes place and where traffic overload is directly experienced.

B. Programmable Network and Virtual Queues

The creation of programmable network devices, such as P4-enabled switches, allows users to customize intelligent packet processing at the data forwarding plane. Users here are the vendors responsible for ensuring QoS to customers of different applications. Some researchers develop virtual queues at the programmable switches to control congestion. Harkous et al. [9] leveraged virtual queues in the P4 pipeline to individually rate-limit each slice. They briefly explored virtual queue-based Active Queue Management (AQM) for congestion policing and meeting latency requirements of different network slices (representing a flow or set of flows). While authors in [10] designed Virtual Dual Queue Core Stateless AQM (VDQ-CSAQM) that depends on two virtual queues to fulfil the goals of Low-Latency Low-Loss Scalable-Throughput (L4S) Internet service and maintain in-network control of resource sharing. These research work suggest AQM techniques using virtual queues, but their work does not serve tactile application. Therefore, similar to these works, we implement virtual queues at the switch/router to differentiate flows into slices and customize them for tactile application.

IV. Proposed System Design and Evaluation

As shown in Fig. 1, all the traffic from our local network passes through the gateway (router) located at the edge before entering other networks (Internet). Thus, the probability of this router being a congestion point is high. Our P4-enabled router is programmed to categorize flows from different applications and separate them into virtual queues (or slices). Each slice has to maintain the respective QoS of the traffic it serves. Table I displays the QoS constraints of the various media streams considered in our framework, and from there, we can observe the time-criticality of haptic streams.

TABLE I
QoS of different application streams [11]–[15]

QoS	Haptic	Audio	Video
Delay (ms)	$\leq$50	$\leq$150	$\leq$400
Jitter (ms)	$\leq$2	$\leq$30	$\leq$30
Packet size (bytes)	64-128	160-320	$\leq$MTU
Packet loss (%)	$\leq$10	$\leq$1	$\leq$1
Update Rate (Hz)	$\geq$1000	$\geq$50	$\geq$30
Throughput (Kbps)	512-1024	64-128	2500-40000

One of the most crucial QoS parameters for us is the end-to-end delay, whose value is dependent on propagation time, transmission time, queuing time, and switch processing time of the packet. We focus on manipulating the egress queues of the router to influence the queuing time and thereby decrease the end-to-end latency. State-of-the-art AQM technique like CoDel is ineffective in meeting the stringent delay requirements of haptic applications when the network is overburdened with heavy data traffic. We thus customize the queue management at each slice to meet the QoS needs and prioritize haptic teleoperation. We introduce the following technical concepts integral to our algorithm.

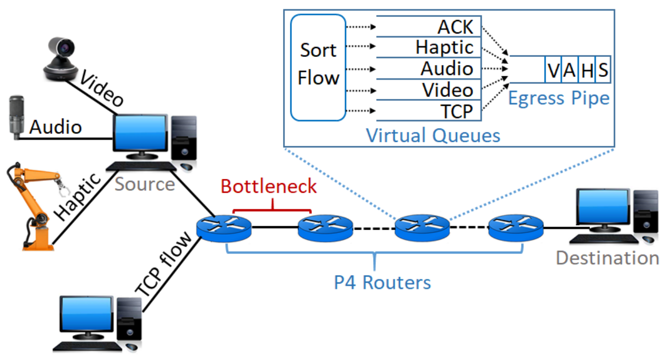

Fig. 2. Tentative framework of the proposed algorithm

Virtual queues as slices: If the packets arriving at the actual queue are serviced by a link with a virtual capacity slightly lower than the outgoing link, the senders would be tricked into filling just the virtual link capacity. We can achieve this by using a virtual queue (VQ) to model the length of a queue. As a result, the link will remain (little) underutilized, and we will be able to maintain a small queue [16]. One straightforward way to design VQ can be increasing its size equal to the packet enqueued at the real queue while decreasing its length slightly lower than the packet dequeued.

As in [9], [10] our proposed algorithm divides different types of network traffic and feeds them into their respective virtual queues. To visualize our system, a tentative framework is shown in Fig. 2.

Classifying traffic: The initial step is to classify the different traffic flows so that they can be directed to the appropriate queues. We assume that different sets of flows correspond to different slices and customize the queue at each slice. First, each flow is distinguished using its 4-tuple attributes (source IP address, source TCP/UDP port, destination IP address, destination TCP/UDP port) and allocated a slice id by the network element (router/switch). Then we classify each traffic and insert them into different priority queues as shown in Fig. 2. We designate the highest priority to the signaling (e.g., for acknowledgment) traffic to rapidly inform TCP senders about network congestion and provide them with appropriate metadata (for INT-driven congestion control) to accordingly pace their traffic. The next priority is given to latency-sensitive haptic traffic, after which we prioritize audio and then the video traffic (which is bandwidth-intensive and latency-sensitive but not as time-critical as haptic and audio signals). We appoint the lowest priority to the TCP background flow generated by non-haptic users.

Dynamic threshold for queues: As per adaptive sampling, if we allocate a fixed bandwidth of time-average data rate to the haptic traffic, we can violate its instantaneous requirements. On the other hand, reserving a bandwidth equal to the peak haptic data rate nullifies the work of adaptive sampling. So we need to dynamically regulate how much bandwidth we can assign for haptic applications without compromising other flows.

We propose a design that can dynamically adjust the size of priority queues allowing full link utilization by lower priority traffic in the absence of haptic samples but immediately providing enough space to handle the unexpected flow of haptic signals.

A. *Evaluation Methodology*

We evaluate our algorithm by implementing it using the BMv2 software switch on the mininet environment. As illustrated in Fig. 2, the haptic feedback is transmitted along with the audio and video streams. In the background of the haptic teleoperation, we run TCP flows. With the increasing number of flows, the switch's offered bandwidth capacity will reach its limit and cause a state of congestion. In this scenario, we can gauge the effectiveness of our solution and compare it against the adaptive sampling method, where we allocate the time-average data rate as bandwidth for haptic traffic. As a QoS parameter, we contrast for each flow its delay measurements generated from our design to that of conventional adaptive sampling and assess the impact of their respective performance during normal conditions and under network congestion.

V. Summary and Future Work

QoS compliance of tactile Internet is not an easy task since the Internet is a complex public network shared between numerous users with various types of traffic. With the help of programmable network elements, we can customize data plane protocols to satisfy QoS requirements even under network congestion. Our proposed algorithm uses these programmable switches to leverage virtual queues whose thresholds we dynamically adapt as per the network conditions. As a future evaluation, we can extend the implementation on a real testbed with hardware programmable network devices like Intel Tofino switch to inspect our algorithm's usefulness and practicability.

Acknowledgment

This work was supported by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy—EXC 2050/1—Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden (Project ID: 390696704) and the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt. " Joint project 6G-life (Project ID: 16KISK001K).

References

[1] V. Gokhale, J. Nair, S. Chaudhuri, and S. Kakade, "Network-aware adaptive sampling for low bitrate telehaptic communication," in *Haptics: Science, Technology, and Applications*, D. Prattichizzo, H. Shinoda, H. Z. Tan, E. Ruffaldi, and A. Frisoli, Eds. Cham: Springer International Publishing, 2018, pp. 660–672.

[2] P4-consortium. (2017, 5) P4 16 Language Specification (version 1.0.0). (accessed 2022-04-07). [Online]. Available: https://p4.org/p4-spec/docs/P4-16-v1.0.0-spec.pdf

[3] 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*, vol. 56, no. 2, pp. 588–597, 2008.

[4] S. Clarke, G. Schillhuber, M. F. Zaeh, and H. Ulbrich, "Telepresence across delayed networks: a combined prediction and compression approach," in *2006 IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE 2006)*, 2006, pp. 171–175.

[5] N. Sakr, N. D. Georganas, and J. Zhao, "Human perception-based data reduction for haptic communication in six-dof telepresence systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3534–3546, 2011.

[6] W. Morita, D. Sugahara, K. Hirata, and M. Yamamoto, "Aqm with multi-queue for microburst in data center networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 379–384.

[7] Y. Yu, X. Jiang, G. Jin, Z. Gao, and P. Li, "A buffer management algorithm based on dynamic marking threshold to restrain microburst in data center network," *Information*, vol. 12, no. 9, 2021. [Online]. Available: https://www.mdpi.com/2078-2489/12/9/369

[8] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh, and M. Yu, "Hpcc: High precision congestion control," in *Proceedings of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 44–58. [Online]. Available: https://doi.org/10.1145/3341302.3342085

[9] H. Harkous, C. Papagianni, K. De Schepper, M. Jarschel, M. Dimolianis, and R. Pries, "Virtual queues for p4: A poor man's programmable traffic manager," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 2860–2872, 2021.

[10] S. Nádas, G. Gombos, F. Fejes, and S. Laki, "A congestion control independent l4s scheduler," in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 45–51. [Online]. Available: https://doi.org/10.1145/3404868.3406669

[11] M. Eid, J. Cha, and A. El Saddik, "Admux: An adaptive multiplexer for haptic–audio–visual data communication," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 1, pp. 21–31, 2011.

[12] K. Iwata, Y. Ishibashi, N. Fukushima, and S. Sugawara, "Qoe assessment in haptic media, sound, and video transmission: Effect of playout buffering control," *Comput. Entertain.*, vol. 8, no. 2, dec 2011. [Online]. Available: https://doi.org/10.1145/1899687.1899694

[13] N. Suzuki and S. Katsura, "Evaluation of qos in haptic communication based on bilateral control," in *2013 IEEE International Conference on Mechatronics (ICM)*, 2013, pp. 886–891.

[14] E. Isomura, S. Tasaka, and T. Nunome, "A multidimensional qoe monitoring system for audiovisual and haptic interactive ip communications," in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, 2013, pp. 196–202.

[15] A. Hamam and A. E. Saddik, "Toward a mathematical model for quality of experience evaluation of haptic applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 12, pp. 3315–3322, 2013.

[16] B. Briscoe, "The native AQM for L4S traffic," *CoRR*, vol. abs/1904.07079, 2019. [Online]. Available: http://arxiv.org/abs/1904.07079