Teleoperation over a Shared Network: When Does it Work?

Vineet Gokhale[1, 2], Jayakrishnan Nair[1], and Subhasis Chaudhuri[1]

[1]Indian Institute of Technology Bombay, Mumbai, India. Email: {vineet, jayakrishnan.nair, sc}@ee.iitb.ac.in
[2]University of South Bohemia České Budějovice, Czech Republic. Email: vgokhale@prf.jcu.cz

***Abstract*— The development of telehaptic communication protocols has gained widespread focus over the past decade. Several protocols have been proposed for carrying out telehaptic activities over a shared network. However, a comprehensive analysis of the impact of the network cross-traffic on the telehaptic stream, and the feasibility of Quality of Service (QoS) compliance, is lacking in the literature. In this paper, we explore the interplay between the telehaptic stream produced by two classes of telehaptic protocols and the cross-traffic that is commonly present in a shared network. Based on this, we formulate general conditions for QoS compliance of the telehaptic stream. These conditions provide guidelines for the design of telehaptic protocols, as well as for the configuration of shared networks for ensuring QoS-compliant telehaptic communication.**

Keywords: *Teleoperation, shared network, QoS compliance*

I. INTRODUCTION

The past two decades have witnessed rapid advancements in the science of exploration and manipulation of remote objects involving the modality of touch – a field generally referred to as *telehaptics*. The primary aim of telehaptics is to provide an immersive environment to the human user for efficiently controlling remote objects through force feedback. Typically, this requires communication of haptic, audio and video information over a network with ultra low latency. Specifically, for seamless telehaptic interaction, stringent Quality of Service (QoS) constraints need to be satisfied for each media type. Table I summarizes the aforementioned specifications for three QoS metrics: frame delay, jitter, and data loss [1].

Media	Delay (ms)	Jitter (ms)	Loss (%)
Haptic	30	10	10
Audio	150	30	1
Video	400	30	1

TABLE I: QoS specifications for media in terms of delay, jitter, and loss for effective telehaptic communication.

In general, non-conformance to the above QoS requirements results in severe perceptual artifacts, thereby causing deteriorated perception of the remote environment. Specifically, violating the haptic QoS constraints destabilizes the global control loop leading to catastrophic effects on the application. Thus, QoS adherence plays a crucial role for a smooth teleoperation. However, on a shared network like the internet, utilized simultaneously by several traffic streams, ensuring QoS compliance poses several challenges. This is because the cross-traffic encountered by the telehaptic stream is both unknown as well as time-varying.

Several protocols have been devised specifically for telehaptic communication over shared networks [2]–[7]. However, performance evaluation of the above mentioned protocols has only been carried out in highly controlled and simplistic network settings. Typically, either no cross-traffic or only constant bit rate (CBR) cross-traffic is considered in the evaluation of these protocols. However, in real-world networks, a majority of the traffic is comprised of Transmission Control Protocol (TCP) flows [8], [9]. Thus, the evaluation of any telehaptic protocol is incomplete without analyzing its interplay with TCP cross-traffic.

This paper seeks to fill this gap. In this paper, we provide a comprehensive assessment of the interplay between telehaptic traffic and heterogeneous cross-traffic, consisting of constant bit rate (CBR) as well as TCP flows. This leads to the formulation of a set of general conditions for guaranteeing telehaptic QoS compliance. Specifically, we focus on the following two classes of telehaptic protocols.

CBR based telehaptic protocols: This class of protocols generates a CBR data stream, i.e., they inject traffic into the network at a steady rate. Examples of such protocols include the Application Layer Protocol for HAptic Networking (ALPHAN) [2], Adaptive Multiplexer (AdMux) [3], Haptics over Internet Protocol (HoIP) [4], and the protocol proposed in [5].

Adaptive sampling based telehaptic protocols: This class of protocols employs adaptive sampling to compress the haptic signal [10]–[13]. The idea behind adaptive sampling is to identify *perceptually significant* haptic samples; transmitting only these samples leads to a substantial reduction in the average telehaptic data rate. Several papers propose using adaptive sampling for telehaptic communication; see, for example, [6], [14]–[17].

For the above classes of protocol, we investigate the interplay of the telehaptic stream and heterogeneous cross-traffic. Our contributions are the following.

1) We perform an analytical characterization of the maximum and minimum end-to-end delay when a CBR-based telehaptic protocol co-exists with a TCP and CBR cross-traffic on a bottleneck link (see Section III-B.1). Further, this analysis also leads to the characterization of the peak haptic jitter (see Section III-B.2).
2) We develop a set of conditions for haptic QoS compliance for CBR-based telehaptic protocols (see Section IV) and adaptive sampling based telehaptic protocols (see Section V). We show that satisfying the

delay and jitter constraints requires the network and cross-traffic parameters to satisfy certain conditions, whereas packet loss depends heavily on the sizes of the telehaptic packets relative to TCP packets. Moreover, we show that the statistical compression achieved by adaptive sampling provides no meaningful economies from the standpoint of network capacity requirements.

II. Typical telehaptic environment

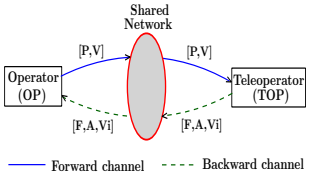

Fig. 1: Framework of a point-to-point telehaptic communication. Data notations: [P,V] - [position, velocity], [F,A,Vi] - [force, audio, video].

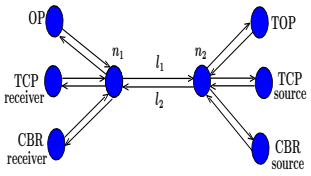

Fig. 2: Network topology showing l_1 and l_2- the bottleneck links in the simulated network; n_1 and n_2 - intermediate nodes.

In this section, we describe the framework of a typical point-to-point telehaptic communication system on a shared network (see Figure 1). The human operator (OP) controls the remote robotic manipulator known as the teleoperator (TOP). The OP transmits the current position and velocity commands on the *forward channel*. The TOP follows the trajectory of the OP through execution of the received commands, and in response it transmits the captured audio and video signals along with the haptic feedback on the *backward channel*.

III. TCP: Background and Analysis

TCP forms the backbone of a wide range of internet applications that demand reliable data transfer, such as web browsing, email, file download, and even video streaming applications like YouTube and Netflix. Studies show that TCP traffic constitutes around 90% of all internet traffic [8], [9]. TCP is a transport layer protocol that controls the rate at which the application injects traffic into the network based on the perceived network conditions. It achieves the end-to-end reliability through retransmission of lost packets, which are detected using packet acknowledgments (ACKs) that are sent to the source by the receiver. In this section, we give a brief overview of TCP NewReno [18], which is the most widely deployed variant of TCP, and provide an analytical characterization of the delay and the jitter encountered by a CBR stream co-existing with a TCP stream on a single bottleneck link. This analysis will be useful when we consider the interplay between TCP and telehaptic protocols that generate CBR traffic (see Section IV).

A. TCP background

We now provide a brief overview of the TCP rate adaptation mechanism. The TCP source maintains a variable called *congestion window* (denoted by W) that defines the number of TCP packets that have been transmitted but not yet acknowledged. The congestion window W controls the rate at which data is injected into the network – a higher W corresponds to a higher transmission rate. Until a packet loss is detected, a TCP source increments W by 1 every round trip time (RTT). This phase is commonly referred to as *congestion avoidance* in literature. Once a packet loss is detected, TCP infers that the network is overloaded and cuts its transmission rate aggressively. This phase is referred to as *fast retransmit, fast recovery* in literature, wherein the TCP source retransmits the lost packet and awaits the ACK corresponding to its reception. Once this ACK is received, the TCP source re-enters the congestion avoidance phase with an initial congestion window that is half the window size at the time the loss was detected.[1]

To provide a concrete visualization, consider the single bottleneck network topology shown in Figure 2 with a single TCP source (we ignore the CBR streams for now). Let μ denote the capacity of the bottleneck link l_2, and B denote the queue size at the ingress of the bottleneck link (i.e. at n_2). Let τ denote the one-way propagation delay of the TCP flow. In this setting, it can be shown that W and the queue occupancy at n_2, denoted by Q, exhibit a cyclic (periodic) variation as shown in Figure 3a (see [19] for details). The interval between t_1 and t_2 corresponds to the congestion avoidance phase. Note that W is incremented in steps of 1, and the resulting increase in transmission rate causes the queue occupancy to increase. Once a packet loss (due to queue overflow) is detected (at time t_2), the source enters the fast retransmit, fast recovery phase. In this phase, which corresponds to the interval between t_2 and t_3 in Figure 3a, the source cuts its transmission rate, causing the queue to drain. Once the source receives the ACK corresponding to the lost packet (at time t_3), it re-enters the congestion avoidance phase and the cycle repeats.

Let W_{min} and Q_{min} denote the minimum value of W and Q in a cycle, respectively. The reference [19] provides an analytical characterization of W_{min} and Q_{min}. Specifically, it is proved in [19] that if $B > 2\mu\tau$,

$$Q_{min} = \frac{B - 2\mu\tau}{2}, \qquad W_{min} = \frac{B + 2\mu\tau}{2S_{tcp}},$$

where S_{tcp} is the size of a TCP packet. In the following section, we generalize the analysis in [19] to include a CBR flow co-existing with the TCP flow on the bottleneck link. This non-trivial analysis leads to a characterization of (i) the maximum and minimum end-to-end CBR delay, (ii) the peak CBR jitter. This characterization will be useful in our subsequent analysis of the interplay between heterogeneous cross-traffic and telehaptic protocols that generate CBR traffic.

B. TCP-CBR interplay

1) Haptic delay characterization: For our analysis, we consider the setting in which there are two traffic flows on the network: a TCP flow and a CBR flow, both transmitting on the bottleneck link l_2. Note that the CBR flow may be composed of multiple CBR connections, including telehaptic

[1]This description assumes a single loss; the congestion window dynamics are more complicated if there are multiple packet losses [18].

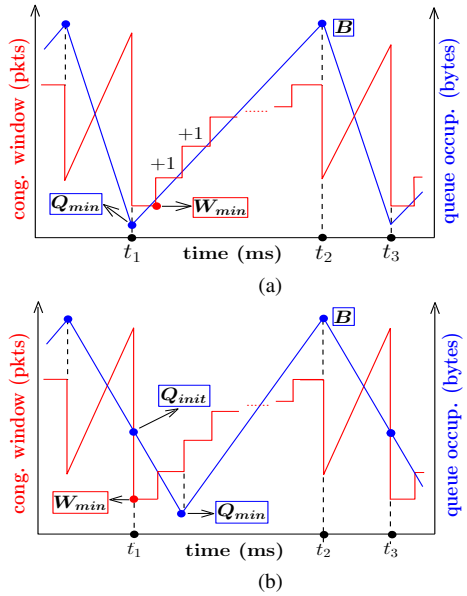

Fig. 3: Evolution of TCP congestion window and bottleneck queue occupancy for (a) single TCP flow (b) heterogeneous traffic flows.

streams. Let R denote the data rate of the CBR flow. For simplicity, we assume that the reverse channel (i.e., link l_1) is uncongested. Moreover, we assume that $B > 2\mu\tau$.

Similar to the case of a single TCP flow, it can be shown that in steady state both W and Q vary periodically in time; see Figure 3b. However, the presence of the CBR cross-traffic changes the nature of the queue occupancy evolution relative to the congestion window evolution. Note that during the congestion avoidance phase (the interval between time t_1 and t_2), Q first decreases and then increases until the queue overflows (i.e. $Q = B$). This results in a shift in the Q_{min} point relative to the W_{min} point.

We now summarize the results of our analysis; we omit the details due to space constraints. Further details of the analysis can be found in [20]. Let Q_{init} denote the buffer occupancy at the start of the congestion avoidance phase. It can be shown that

$$Q_{init} = \frac{B + 2\mu\tau}{2}\left(1 + \frac{R}{\mu}\right) - 2\mu\tau,$$
$$W_{min} = \frac{B + 2\mu\tau}{2S_{tcp}}\left(1 - \frac{R}{\mu}\right).$$

However, Q_{min} does not admit a closed form characterization. We develop the following numerical procedure for computing Q_{min}. Taking $\alpha = \frac{R}{\mu}$, the following equation describes the evolution of the queue occupancy after i updates in W, starting from t_1. Note that $W(1) = W_{min}$ and $Q(1) = Q_{init}$.

$$Q(i) = Q_{init}\alpha^{i-1} + \left(2\mu\tau + \frac{W_{min}S_{tcp}}{1-\alpha}\right)(1 - \alpha^{i-1}) + \sum_{j=0}^{i-3}(i - 2 - j)\alpha^j.$$

Q_{min} can be computed by minimizing $Q(i)$ over i.

The above results enable a characterization of the minimum and maximum end-to-end delays experienced by TCP/CBR packets. The minimum delay d_{min} and the maximum delay d_{max} are experienced when the queue occupancy is at its minimum (Q_{min}) and maximum (B), respectively, and are given by

$$d_{min} = \tau + \frac{Q_{min}}{\mu}, \qquad d_{max} = \tau + \frac{B}{\mu}.$$

Therefore, the end-to-end packet delay varies cyclically over the range $[d_{min}, d_{max}]$. When the CBR flow includes telehaptic data, these bounds allow us to check if QoS-compliant telehaptic communication is feasible in a given network configuration. We elaborate on this in Section IV.

2) Haptic jitter characterization: Our analysis of the CBR-TCP interplay also leads to a characterization of peak jitter encountered by the haptic samples. To express the peak haptic jitter, we split the CBR stream into two flows: a CBR telehaptic flow of rate R_h, and a CBR cross-traffic flow of rate R_{cross} such that $R = R_h + R_{cross}$. The expression for the peak haptic jitter (denoted by ν_{max}) is given as follows; we omit the details due to space constraints.

$$\nu_{max} = \frac{m_{tcp} + m_{cross}}{\mu} - T_h,$$

where m_{tcp} and m_{cross} denote the maximum volume of TCP and CBR cross-traffic injected into the queue between two successive haptic packets, respectively, and are expressed as follows.

$$m_{tcp} = \left[n + 1 + \left(1 + \left\lfloor \frac{T_h - \frac{nS_{tcp}}{\mu}}{\frac{nS_{tcp}}{\mu - R}} \right\rfloor\right) nI_{\left(T_h > \frac{nS_{tcp}}{\mu}\right)}\right] S_{tcp}$$
$$m_{cross} = \left(\left\lceil \frac{R_{cross}T_h}{S_{cross}} \right\rceil + 1\right) S_{cross}$$

Here, n denotes the cumulative acknowledgement parameter of the TCP flow, T_h denotes the inter packet gap of the telehaptic flow, $I_z = 1$ if $z = 1$ and 0 otherwise. S_{cross} denotes the packet size of the CBR cross-traffic flow. Further details of the peak haptic jitter characterization is available in [20].

IV. CBR-Based Telehaptic Protocols

In this section, we validate our analysis of the interplay between CBR-based telehaptic traffic and heterogeneous cross-traffic consisting of TCP as well as CBR flows. Further, we develop an understanding of the conditions that need to be satisfied for ensuring QoS-compliant telehaptic communication on a shared network for CBR-based telehaptic protocols.

For our experiments, we use NS3 – a discrete event network simulator [21]. We consider the single bottleneck network topology as shown in Figure 2. Unless otherwise specified, we use the following network settings throughout this section. We set $\mu = 6$ Mbps, $\tau = 8$ ms and $B = 14$ kB.[2] We work with real world haptic traces generated by the Phantom Omni device [22], which offers a single

[2]The chosen settings represent a medium speed internet link of length approximately equal to 1000 miles.

point of interaction between the human user and the haptic environment. Considering the standard haptic sampling rate of 1 kHz, and accounting for the overhead due to packet headers, we get a forward channel telehaptic data rate $R_f =$ 688 kbps, with packets of size 86 bytes transmitted every millisecond ($T_h = 1$ ms) [7]. On the backward channel, we simulate audio and video payload at the rate of 64 kbps and 400 kbps, respectively. We consider the media multiplexing mechanism proposed in [7], where each packet contains a single haptic sample and an audio/video fragment of fixed size. Including packet headers, this leads to a backward channe telehaptic data rate of $R_b = 1.096$ Mbps, with packets of size 137 bytes transmitted every millisecond.

For brevity, we report the simulation results for the case in which the cross-traffic sources are added to the backward channel only. For TCP traffic, we use a TCP NewReno source with the standard packet size $S_{tcp} = 578$ bytes, and $n = 2$. We also add a CBR cross-traffic source with data rate R_{cross} (used as a control parameter), and packet size $S_{cross} = 150$ bytes.[3] In the notation of Section III-B, note that the aggregate CBR rate on the backward channel $R = R_b + R_{cross}$. For sustaining the TCP flow throughout the duration of the experiment, we need to ensure that $R < \mu$ so that the TCP flow has sufficient network bandwidth to perform rate adaptation. Our simulations are performed for a duration of 500 seconds. Due to space limitations, and given that haptic QoS requirements are stricter than those for audio/video, we only report haptic QoS measurements here.

We now report our results corresponding to the three QoS metrics: delay, jitter, and packet loss.

A. Delay

Through simulations, we note that the TCP flow does not remain in steady state for $R > 5.5$ Mbps. Hence, we restrict our measurements to a maximum $R = 5.5$ Mbps. We validate the analytical bounds (derived in Section III) as well as the measured minimum and maximum haptic delay by varying R_{cross} to get R in the range [R_b, 5.5 Mbps]. We note that throughout the range of R, our analytical model gives accurate estimates of the minimum and maximum haptic delays. In Figure 4, we plot the temporal variation of the haptic delay for $R = 3$ Mbps (i.e., $R_{cross} = 1.904$ Mbps), along with the analytical bounds. As expected, the haptic delay evolves periodically in time. We make the following remarks.

- The upper bound d_{max}, which is insensitive to R, remains highly accurate. However, we observe in our traces that for large values of R, TCP suffers multiple losses per cycle, leading to a very different congestion window evolution from the one analysed. Simulating a wide range of network settings, we observe that a sufficient condition for a single TCP packet loss per cycle (and consequently for the accuracy of d_{min}) is $R \leq 0.65\mu$.

[3]This is the typical packet size of a video-conferencing application such as Skype.

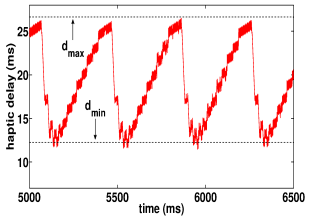

Fig. 4: Plot of haptic delay demonstrating the corroboration between theoretical estimates and simulation measurements.

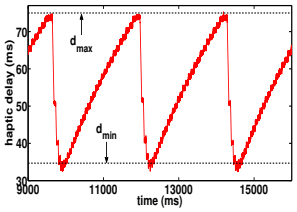

Fig. 5: Plot of haptic delay with analytical and simulation bounds demonstrating the severe violations of haptic delay QoS constraints.

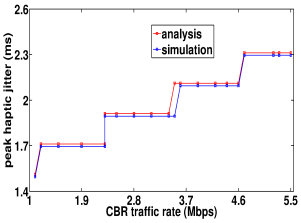

Fig. 6: Plot showing the haptic jitter profile for Phantom Omni haptic device.

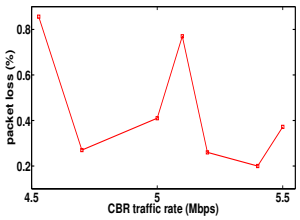

Fig. 7: Packet loss on the backward channel for a higher dimensional haptic device.

- Since the analytical upper bound d_{max} is highly accurate, it can be used to check for compliance of the haptic delay constraint for a given network setting. In the network setting under consideration, $d_{max} = 26.66$ ms which is less than the QoS limit of 30 ms. Indeed, our measurements confirm that the haptic delay constraint is satisfied in this case.
  To see another example, consider the setting $\mu = 6$ Mbps, $\tau = 15$ ms, and $B = 45$ kB. In this case, $d_{max} =$ 75 ms which suggests that the haptic delay constraint cannot be met. Indeed, simulations show that this is the case; see Figure 5.

Hence, one needs to appropriately configure the network parameters (B, μ and τ) to ensure $d_{max} < 30$ ms.

B. Jitter

We now move to validation of our peak haptic jitter characterization. Figure 6 shows the haptic jitter curves, by both analysis and simulations, as functions of R. It can be seen that the peak haptic jitter follows a piecewise constant profile in R, thereby validating our analytical model. For this setting, the peak jitter remains comfortably within the QoS limit of 10 ms. As another example, consider $n = 5, S_{tcp} =$ 1.5 kB. For the setting, we get $\nu_{max} = 12$ ms, which clearly violates the jitter deadline. Hence, in a given network setting, in order to adhere to haptic jitter QoS requirements, one needs to appropriately configure the cross-traffic parameters (n, S_{tcp}, S_{cross}) such that $\nu_{max} < 10$ ms.

C. Packet loss

Finally, we turn our focus on the packet losses suffered by the telehaptic stream. Interestingly, we notice that telehaptic packet losses are *zero* in spite of the regular queue overflows induced by TCP. The reason for this is that the telehaptic

stream uses smaller packets compared to TCP (137 bytes per packet for the telehaptic stream, versus 578 bytes per packet for the TCP stream). As a result, even when the queue drops a TCP packet, the adjacent telehaptic packets can still (potentially) be accommodated.

To confirm our conjecture that smaller telehaptic packet sizes are responsible for the absence of telehaptic packet losses, we simulate a scenario with higher resolution haptic, audio, and video devices, so that the telehaptic packet size becomes comparable to the TCP packet size. Specifically, consider a haptic device like Cybergrasp [23] or Festo's exohand [24]. Assuming two interaction points for each of the ten fingers of the hands results in a twenty-fold increase in the haptic payload rate. Additionally, we simulate a high-definition audio and video payload with rates of 128 kbps and 2 Mbps, respectively. This results in $R_b = 4.528$ Mbps, and a packet size of 566 bytes on the backward channel for every millisecond. Figure 7 presents the packet loss (in %) encountered by this telehaptic stream, where we vary R_{cross} to get R in the range [R_b, 5.5 Mbps]. Note that with the larger telehaptic packets, losses do occur. While the measured telehaptic losses pose no threat to haptic media with a QoS limit of 10%, the audio and video, which have a more stringent limit of 1%, are susceptible to QoS violations.

To summarize, if telehaptic packets are small relative to TCP packets, the telehaptic stream sees little or no packet loss. However, if the telehaptic packets become comparable in size to TCP packets (due to higher fidelity media devices), packet losses become noticeable.

In conclusion, we see that for QoS-compliant communication under CBR-based telehaptic protocols, the following conditions need to be satisfied.

- For buffer stability, we naturally require that the aggregate CBR data rate is less than the link capacity, i.e., $R < \mu$.
- In order to satisfy the haptic delay constraint, we need $d_{max} < 30$ ms, i.e.,

$$\tau + \frac{B}{\mu} < 30. \quad (1)$$

- In order to satisfy haptic jitter constraint, we need $\nu_{max} < 10$ ms.
- To avoid loss in the presence of concurrent TCP traffic, the packet sizes used by the telehaptic protocol should be small compared to the TCP packet sizes.

V. Adaptive Sampling based Telehaptic Protocols

In this section, we seek to study the interplay between adaptive sampling based telehaptic traffic and heterogeneous cross-traffic involving TCP and CBR flows. An adaptive sampling based protocol transmits only perceptually significant haptic samples on the forward and/or backward channels [6], [14]–[16]. As before, the goal of this section is to understand the conditions for QoS-compliant telehaptic communication.

If the protocol employs Weber sampling [10] on the backward channel, it must also specify how the perceptually

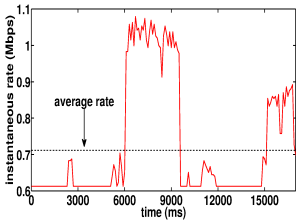

Fig. 8: Instantaneous telehaptic rate exhibiting rapid fluctuations under visual-haptic multiplexing.

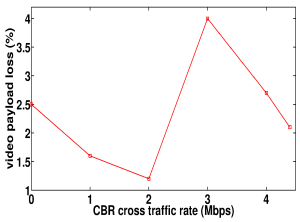

Fig. 9: Video payload loss in presence of heterogeneous cross-traffic under visual-haptic multiplexing.

significant haptic samples are multiplexed with audio/video data. For a working example, we consider the visual-haptic multiplexing protocol [6], which assumes haptic and video data on the backward channel. It multiplexes these two media streams as follows: The perceptually significant haptic samples are packetized with video data of worth 1 ms. On the other hand, when a series of haptic samples are perceptually insignificant, the protocol packs a large chunk of a video frame, not exceeding data of worth 15 ms, into a single packet.

In order to evaluate this protocol with realistic haptic data, we record ten pilot signals collected from Phantom Omni device during a real telehaptic activity. For brevity, we report results only for one of these traces, but we note that our findings apply to the remaining traces as well. The video payload rate is set to 400 kbps, as before. We use the network settings described previously in Section IV. In Figure 8, we plot the instantaneous telehaptic transmission rate on the backward channel due to this protocol. It can be seen that the instantaneous rate exhibits large fluctuations within the range [613, 1079] kbps, while the long term average rate of 712 kbps is substantially lower compared to the peak instantaneous rate (1079 kbps). We now turn to the interplay between this telehaptic flow and network cross-traffic. We begin by considering the impact of CBR cross-traffic alone, and then move to the heterogeneous cross-traffic case.

A. CBR cross-traffic

In this section, we consider the interplay between the visual-haptic multiplexing protocol and CBR cross-traffic on the backward channel. Note that the TCP source is turned off for this simulation. Our goal is to demonstrate that from the standpoint of QoS compliance on a shared network, the statistical compression provided by adaptive sampling is not particularly useful. In other words, the network has to be able to support the *peak* transmission rate of the telehaptic flow for QoS compliance.

To see this, we consider an example where the network is provisioned for the average telehaptic rate. To simulate this scenario, we set $R_{cross} = 5.28$ Mbps so that the bandwidth available to the telehaptic stream is 720 kbps (larger than average rate of 712 kbps). In Figure 8, consider the interval between 6000 ms and 10000 ms, when the instantaneous rate exceeds the available bandwidth. In this interval, our simulation traces reveal a significant haptic

and video payload losses of around 6.2%. Even though the haptic loss is below the QoS limits (10%), the video loss is alarmingly high, causing severe violations of the QoS requirement (1%). As another example, setting $\mu = 3$ Mbps and $R_{cross} = 2.28$ Mbps results in larger haptic and video payload losses of around 9.6%.

B. Heterogeneous cross-traffic

For the case of heterogeneous cross-traffic, we reinstate the TCP source on the backward channel. Since $R_b \in$ [613, 1079] kbps, we vary R_{cross} in the range [0, 4.4] Mbps so that $R \in [R_b$, 5.5 Mbps]. Due to space limitations, we omit the results for haptic delay, noting that Equation (1) remains the condition for meeting the delay constraint. Figure 9 shows the video payload loss (in %) recorded for a wide range of R. It can be seen that the loss for video faces severe QoS violations irrespective of the R value. This is because the visual-haptic multiplexing protocol transmits large packets with video payload in the absence of perceptually significant haptic samples. Recall, from our discussion in Section IV-C, that large packets are more likely to get dropped in the presence of TCP cross-traffic. Interestingly, haptic media suffers *zero* losses in this case. This is because the protocol transmits the perceptually significant samples in smaller packets of size 137 bytes. Once again, this illustrates that the packet sizing performed by a telehaptic protocol plays a crucial role in the loss experienced in the presence of TCP cross-traffic.

To summarize, the conditions for QoS compliance for adaptive sampling based telehaptic protocols are:

- The network be provisioned for the peak telehaptic rate as telehaptic applications are highly sensitive to QoS violations.
- To meet the haptic delay constraint, Equation (1) be satisfied.
- To avoid packet loss in presence of a TCP flow, large packet sizes be avoided.

VI. Concluding remarks

In this paper, we presented a comprehensive assessment of the interplay between telehaptic protocols and heterogeneous cross-traffic in a shared network. We characterized bounds on the haptic delays and jitter through the design of a mathematical model whose accuracy was validated through extensive simulations. Using the expressions for maximum haptic delay and jitter, we derived sufficient conditions for meeting the haptic delay and jitter QoS deadline in terms of network and cross-traffic parameters. Additionally, we derived two important, simulation-driven conditions for QoS compliance. First, in order to ensure a near-zero packet loss the telehaptic packets need to be small relative to the TCP packets. Second, when employing the adaptive sampling for telehaptic communication the network should be provisioned for the peak telehaptic rate to prevent QoS violations.

References

[1] A. Marshall, K. M. Yap, and W. Yu, "Providing qos for networked peers in distributed haptic virtual environments," *Advances in Multimedia*, 2008.

[2] H. Al Osman, M. Eid, R. Iglesias, and A. El Saddik, "Alphan: Application layer protocol for haptic networking," in *Haptic, Audio and Visual Environments and Games, 2007. HAVE 2007. IEEE International Workshop on*. IEEE, 2007, pp. 96–101.

[3] 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.

[4] V. Gokhale, S. Chaudhuri, and O. Dabeer, "HoIP: A point-to-point haptic data communication protocol and its evaluation," in *Twenty First National Conference on Communications (NCC)*. IEEE, 2015, pp. 1–6.

[5] 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*. ACM, 2005, pp. 1–6.

[6] B. Cizmeci, R. Chaudhari, X. Xu, N. Alt, and E. Steinbach, "A visual-haptic multiplexing scheme for teleoperation over constant-bitrate communication links," in *Haptics: Neuroscience, Devices, Modeling, and Applications*. Springer, 2014, pp. 131–138.

[7] V. Gokhale, J. Nair, and S. Chaudhuri, "Congestion control for network-aware telehaptic communication," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 2, p. 17, 2017.

[8] S. Ryu, C. Rump, and C. Qiao, "Advances in internet congestion control," *IEEE Communications Surveys & Tutorials*, vol. 5, no. 1, pp. 28–39, 2003.

[9] S. Yao, F. Xue, B. Mukherjee, S. B. Yoo, and S. Dixit, "Electrical ingress buffering and traffic aggregation for optical packet switching and their effect on tcp-level performance in optical mesh networks," *IEEE Communications Magazine*, vol. 40, no. 9, pp. 66–72, 2002.

[10] 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.

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

[12] A. Bhardwaj, O. Dabeer, and S. Chaudhuri, "Can we improve over weber sampling of haptic signals?" in *Information Theory and Applications Workshop (ITA), 2013*. IEEE, 2013, pp. 1–6.

[13] 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.

[14] E. Steinbach, S. Hirche, J. Kammerl, I. Vittorias, and R. Chaudhari, "Haptic data compression and communication," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 87–96, 2011.

[15] Q. Nasir and E. Khalil, "Perception based adaptive haptic communication protocol (pahcp)," in *Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on*. IEEE, 2012, pp. 1–6.

[16] V. Gokhale, O. Dabeer, and S. Chaudhuri, "HoIP: Haptics over internet protocol," in *IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE)*, 2013, pp. 45–50.

[17] V. Gokhale, J. Nair, and S. Chaudhuri, "Opportunistic adaptive haptic sampling on forward channel in telehaptic communication," in *Haptics Symposium (HAPTICS)*. IEEE, 2016.

[18] S. Floyd, A. Gurtov, and T. Henderson, "The newreno modification to tcp's fast recovery algorithm," 2004.

[19] J. Sun, M. Zukerman, K.-T. Ko, G. Chen, and S. Chan, "Effect of large buffers on tcp queueing behavior," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2004, pp. 751–761.

[20] V. Gokhale, "Telehaptic communication over shared network: Protocol design and analysis," Ph.D. dissertation, Indian Institute of Technology Bombay, 2017.

[21] ns3, "The network simulator," 2011. [Online]. Available: http://www.nsnam.org/

[22] "Phantom omni device reference: www.sensable.com/haptic-phantom-omni.htm," july 2012.

[23] "Cybergrasp user's guide v1.2, immersion corporation," San Jose, CA, USA, 2003.

[24] "Festo exohand: http://www.festo.com/," 2013.