IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 979-8-3503-8447-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOMWKSHPS61880.2024.10620882

Towards a TSN-DetNet Intercity Testbed for Tactile Cyber-Physical Systems

Joydeep Pal, Deepak Choudhary*, Nithish Krishnabharathi Gnani*,
T V Prabhakar, Chandramani Singh, Hari Krishna Atluri[1], and A. Paventhan[1]
Indian Institute of Science, Bengaluru, India [1] *ERNET India*
{joydeeppal, deepakcl, nithishgnani, tvprabs, chandra}@iisc.ac.in, [1]{hari.krishna, paventhan}@ernet.in

***Abstract*—We present a TSN-DetNet testbed design for Tactile Cyber-Physical Systems (TCPS) that facilitates Time-Sensitive Networking (TSN) and Deterministic Networking (DetNet) standards to provide low-latency and high-reliability communication across two cities separated by 400 kilometres. The core design revolves around two key domains: the CPS domain, encompassing haptic devices, robotic arms, and sensing equipment, and the Tactile Internet domain, comprising TSN-enabled LAN switches and Multiprotocol Label Switching (MPLS)-based WAN routers. This configuration ensures the handling of diverse traffic classes, including Scheduled Traffic (ST) for critical tactile data and Best Effort (BE) for non-critical flows. We leverage Precision Time Protocol (PTP) for time synchronization. The paper comprehensively evaluates the testbed's performance metrics, encompassing latency, jitter, packet loss, and time synchronization accuracy. Subsequently, it showcases the practical capabilities of the testbed through a real-world demonstration of a robotic teleoperation application, incorporating haptic and video feedback functionalities. The paper also discusses the challenges of building an intercity TSN-DetNet testbed and future work for using it to demonstrate and evaluate TCPS applications.**

***Index Terms*—Deterministic Networking (DetNet), Network Measurement, P4, SmartNIC, Testbed, Time-Sensitive Networking (TSN)**

I. Introduction

Tactile Cyber-Physical System (TCPS) applications such as remote robotic surgery lay down stringent latency and reliability requirements for their successful operations. The service requirements for cyber-physical control applications are stated in 3GPP TS 22.104 V19.1.0 (2023-09) [1]. The IEEE 802.1Q Time-Sensitive Networking (TSN) Group is a body of standards, where the IEEE 802.1Qbv (Time-Aware Shaper) [2], IEEE 802.1CB (Frame Replication and Elimination for Reliability) [3], and IEEE 802.1AS (Clock Synchronization) [4] standards are comprehensive to support such applications over Ethernet protocol.

The 3GPP TS [1] specification suggests a maximum end-to-end latency of 20 milliseconds for remote surgery applications for up to 2 User Equipment per $1000km^2$. Furthermore, the Internet Engineering Task Force (IETF) proposes the Deterministic Networking (DetNet) standard [5] to support bounded latency over a Wide Area Network (WAN). A Multi-Protocol Label Switching (MPLS) link is expected to provide gateway services for TSN traffic.

* Equal contributors

In this work, we introduce a custom-built testbed, depicted in Fig. 1, designed to establish an end-to-end TSN-DetNet network spanning a distance of 400 kilometres across two cities. The testbed architecture encompasses switches S2 and S3, providing TSN-DetNet Gateway capabilities. The TSN domain, situated within each city, comprises TSN-enabled Switches S1 and S4, which employ the TAS mechanism to schedule traffic originating from end-host machines h1, h2, h3, and h4. These machines are synchronized using Precision Time Protocol (PTP). The IEEE 802.1AS standard for clock synchronization utilizes the services of PTP. End-host machines h1 and h3 assume the role of Scheduled Traffic (ST) sources, tasked with the transmission and reception of tactile data, thereby integrating CPS components such as haptic devices, VR headset, depth camera, and robotic arm within the testbed. Conversely, end-host machines h2 and h4 generate Best Effort (BE) traffic, representing non-critical network flows.

In this paper, we present the following key contributions:

- We design a TSN-DetNet testbed for TCPS applications between two cities separated by 400 kilometres.
- We perform testbed experiments on feasibility and performance on two traffic classes - Scheduled Traffic (ST) and Best Effort (BE).
- We consider an application of robotic teleoperation and demonstrate the feasibility of our testbed.

II. Related Work

TCPS and Tactile Internet (TI) are an evolving field of study. Therefore, the majority of the current endeavours within this field are confined to a lab environment with network simulations. Kurian et al. [6] develop a tactile testbed in a modular architecture. The network is simulated in ns-3 and the teleoperator is simulated in VREP, a robotic simulation platform. An edge-intelligence module has also been introduced. Arjun et al. [7] develop a tactile testbed with a haptic device and PhantomX teleoperated robotic arm. They characterize latencies for a typical teleoperation performed over a LAN. Gokhale et al. [8] develop a testbed with a haptic device and an emulated network for evaluating haptic applications using a novel haptic protocol and satisfying pre-configured QoS limits. The teleoperator domain is entirely in simulation. In another work, TIXT [9] presents a TI testbed and highlights

979-8-3503-8447-5/24/$31.00 ©2024 IEEE

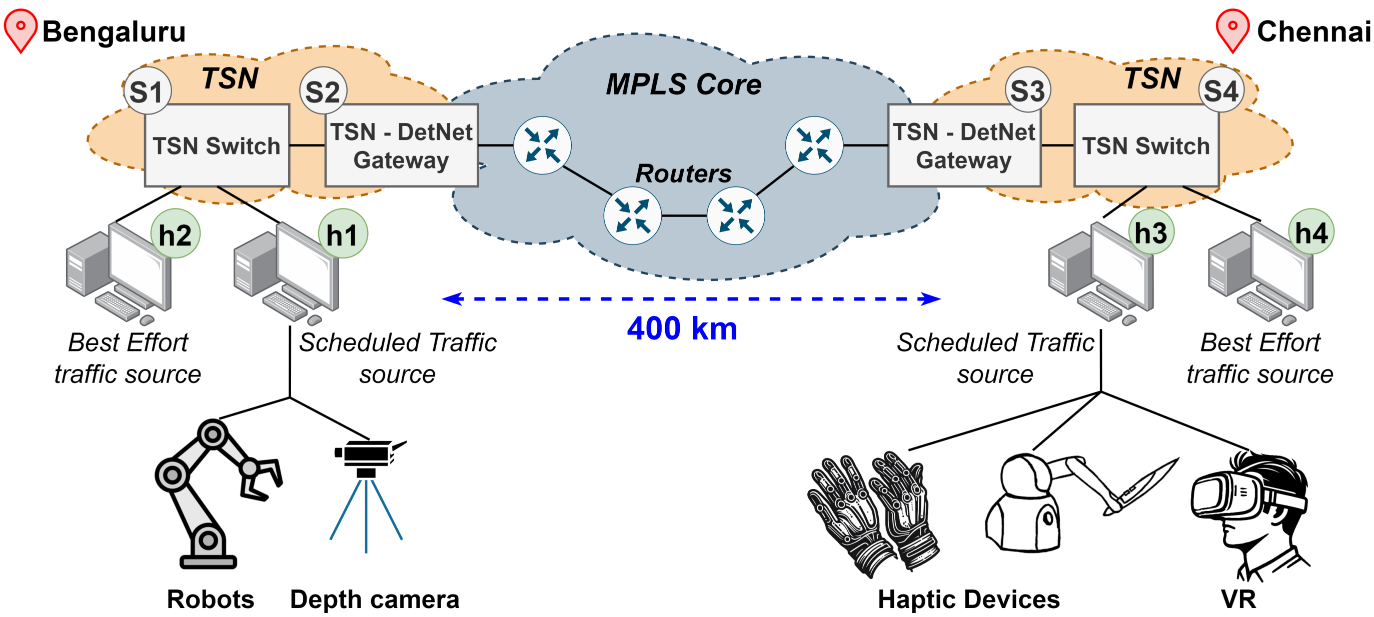

Fig. 1. TSN-DetNet Testbed for Tactile CPS consists of haptic devices, robotic arms, sensing, and feedback equipment connected to TSN switches developed using programmable SmartNICs. A 400 km Wide Area Network between two cities consists of MPLS-capable routers.

challenges in its realization due to the need for close collaboration between diverse disciplines. Simulations with simplified network models may not fully capture real-world complexities. To address this gap, our testbed incorporates a real intercity network alongside CPS hardware to test and evaluate TCPS applications. It also allows us to program complete networking behaviour i.e. Layer 2 and Layer 3, using programmable SmartNICs as our networking switches and routers.

III. Testbed Building Blocks

We present our custom-built TSN-DetNet testbed for TCPS applications in Fig. 1. This testbed consists of two domains: the CPS domain and the Tactile Internet domain. The Operator domain encompasses all the instances that generate traffic into the network. The traffic transmitted to the network can belong to either ST flows or BE flows.

A. CPS Domain

1) Human-operator side: The human operator interacts with real and virtual devices and the environment with the means of haptic devices and visualization equipment. This includes two Geomagic Touch haptic devices [10], a pair of SenseGlove Nova haptic gloves, and two Meta Quest 2 VR headsets [11].

2) Teleoperator side: The testbed has a UR3 robotic arm (CB version) [12], equipped with a Robotiq 2F-85 two-finger gripper [13] and a Robotiq FT 300-S force-torque sensor [14]. A custom-built force sensor array is attached to the gripper for contact area sensing. An Intel RealSense D415 depth camera [15] is used for depth sensing in the teleoperator environment. This robotic system is capable of carrying out precise teleoperation tasks with accurate sensing capability.

B. Tactile Internet Domain

The experimental testbed utilized in this paper is illustrated in Fig. 1. The Tactile Internet domain is designed to facilitate the bidirectional transmission of haptic coordinates and feedback between the operator and teleoperator ends while concurrently supporting other network flows. This network supports low round-trip time latency and provides high reliability. The domain's architecture is further subdivided into two subdomains: the TSN Domain, situated in Bengaluru and Chennai, and the DetNet Domain. The TSN Domain consists of end-host machines interconnected via TSN-enabled switches. The end-host machines are equipped with Intel x86 processors with Linux-based operating system (OS) and Network Interface Cards (NICs) that can support data transmission of up to 10Gbps.

Traffic Flows in the TSN Domain can be categorized as Scheduled Traffic (ST) flows and Best Effort (BE) flows. An ST flow is characterized by its stringent latency requirements and guaranteed bandwidth allocation. A BE flow is not subjected to strict timing guarantees. To differentiate between ST and BE flows, TSN utilizes Virtual LAN (VLAN) IDs. Each flow type is assigned a specific VLAN ID, enabling the network to prioritize and manage them effectively. To seamlessly interoperate with the DetNet domain, a TSN-DetNet Gateway is utilized. This is responsible for transforming TSN flows, identified by VLAN IDs, into DetNet flows compatible with MPLS. The VLAN IDs associated with each TSN flow are converted to MPLS flow with a unique MPLS label. The exchange of PTP messages between end-host instances is facilitated using the ST flow link.

In the TSN domain, we have end-hosts that transmit tactile packets with latency and reliability requirements. These packets are part of the ST flow. The intermediate TSN-enabled switches ensure that the ST flow obtains an end-to-end bounded latency. This determinism is achieved by servicing the ST flow and BE flow in alternating gate slots of a Cycle Time (CT). This is configured on all the TSN switches of the testbed.

The DetNet domain comprises routers that leverage MPLS label-based forwarding for ST flow. This approach ensures reliable delivery of ST packets, provided the transmission data rate remains under 50 Mbps. In contrast, BE traffic within the DetNet domain is forwarded through traditional IP-based routing, catering to non-critical network flows.

IV. Testbed Design

A. CPS Design

We design the human operator side and teleoperator side of the testbed for a CPS application described in Section V-C.

1) Human operator design: The Geomagic Touch haptic device has 6 degrees of freedom (DOF) positional sensing and 3 DOF force feedback using its three motors. To be operated for TCPS applications on the human operator side, it was configured to sense the kinematic data of its stylus (position, velocity, joint angles) and send it to a host computer using serial communication. It receives input signals from the host to send actuation signals to the joints of its motors to provide vibrational and/or force feedback to the human operator. It also has two buttons which can be used to send toggle commands. The sensing and actuation is performed at $1\ KHz$. A non-blocking multi-threaded code was developed to get kinematic data from the Geometric Touch using the OpenHaptics toolkit and C programming tools. The position coordinates, and button data are packaged into a standard structure for the teleoperation application and sent as UDP packets using socket programming. A monitor connected to the host is used to display the live video feed of the teleoperator side.

2) Teleoperator design: The UR3 CB industrial collaborative robotic arm is designed to perform tasks that are pre-programmed using Universal Robots' patented programming interface, PolyScope. In its basic configuration, it does not have real-time control capability. Interestingly, it also has the feature to be externally controlled via Ethernet by installing the *External_Control* plugin into PolyScope. By installing the *Universal_Robots_ROS_Driver* in a host computer, we use the Robot Operating System (ROS) to externally control the robot using the network. This open-source driver was made to provide a stable and sustainable interface between UR robots and ROS. A script was written in Python programming using the *rospy* library for real-time control of the robot. The program uses sockets to receive UDP packets from the human operator side and extracts the position data (cartesian coordinates). We make use of the *CartesianTrajectoryController* in ROS to control the robot's motion. The robot thus interpolates to the coordinates commanded by the haptic device. The real-time kinematic data from the robot is read and packaged into the standard structure and sent as feedback via UDP packets.

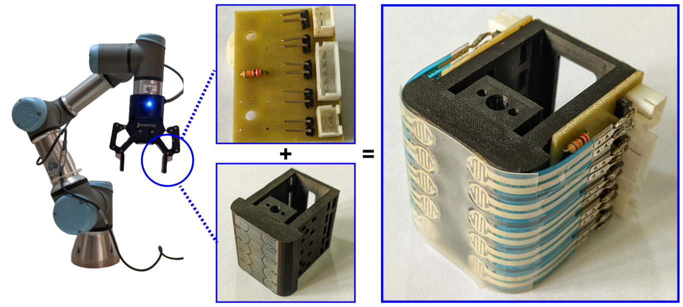

Fig. 2. Enhancing gripping with contact area sensing capability using a force sensor array

3) Contact area sensing: The robot is equipped with a two-finger gripper. While it is capable of controlling the force and position to grasp an object, the real-time orientation of the object to the gripper fingers is unknown. We designed a force sensor array using low-cost Force Sensitive Resistors (FSR) to enable contact area sensing as shown in Fig. 2. It helps in accurate object manipulation. Each FSR is connected to the Arduino Mega2560Rev3 microcontroller via a voltage divider circuit with a $23k\Omega$ pull-down resistor. The FSRs behave as variable resistors between the reference voltage (V_{ref}) and input voltage (V_{in}) measured at the Arduino's 10-bit ADC pins. The resulting contact data from the FSR array is transmitted to the robot host using the *pySerial* serial communication library. This data is added to the kinematic feedback packets from the teleoperator. The custom-built PCB is compact in shape and attaches to a modified finger housing manufactured using 3D printing. The design attaches to the gripper without impeding its regular operation.

B. Tactile Internet Design

The x86 Linux end-host machines are running Ubuntu 22.04.6 LTS OS. Each end-host machine is equipped with an Intel i5-12900 CPU and dual-port Intel X520-DA2 10G NICs, capable of hardware timestamping. The end-host machine is connected to the TSN switch. The connections in the TSN domain are made using OM3 Multi-mode optical fibre cables.

Within the end-host machines, UDP packets encapsulating the tactile data are transmitted through a designated network interface on the NIC, configuring a VLAN ID of 160 for ST flow identification within the TSN domain. Concurrently, BE traffic flow is identified using a distinct VLAN ID of 150. The TSN switch implements the TAS mechanism through the TAPRIO queuing discipline (qdisc). The TAPRIO qdisc uses Linux's Socket Buffer (SKB) priority framework to classify traffic. To achieve this, we establish a mapping between the VLAN tags associated with ST and BE flows and the corresponding SKB priority classes using iptables, a user-space Linux utility. This configuration guarantees the prioritization of ST and BE flows at the egress port, ensuring deterministic delivery of critical tactile data while accommodating less time-sensitive network traffic.

At the intersection of the TSN and DetNet domains, the TSN-DetNet gateway applies MPLS labels to the ST flows, directing them toward the DetNet domain. Within this domain, the routers, configured for label-based forwarding, ensure reliable delivery of ST flow packets, guaranteeing the reliability of time-sensitive tactile traffic. Conversely, the BE flows are forwarded to the routers adhering to the conventional IP-based routing.

To synchronize the end-host machines across a WAN link, we use PTP for precise synchronization. PTP Hardware Clocks (PHCs) on each NIC were synchronized using the Linux utility ptp4l, while the host system time was aligned with the PHCs using the Linux utility phc2sys.

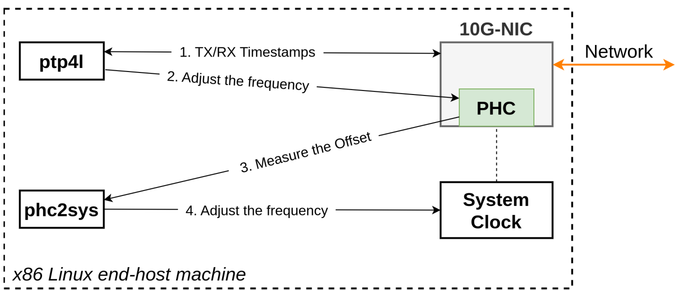

Fig. 3. Synchronizing the System Clock on an end-host machine

While conventional PTP employs multicast communication, the PTP messages cannot traverse subnets across cities with core routers disabling multicast forwarding. To address this challenge, we configured a unicast-based PTP mechanism, enabling precise synchronization of end-host PTP instances residing in separate subnets across a WAN link.

On the end-host machines, we run the leader and follower PTP instances using the custom unicast profile of PTP. We run the unicast profile for end-host synchronization using ptp4l, which begins with exchanging PTP messages and synchronizing with each other. The ptp4l instances on each end-host machine ensure that all the PHCs are synchronized. To synchronize the host system clock with the PHC, phc2sys is used.

V. Testbed Evaluation

We perform comprehensive measurements on the testbed to measure latency, packet loss, and jitter. We also implement PTP on the end-host machines on either side and measure the synchronization accuracy. Finally, we evaluate the performance of a teleoperated drawing task.

A. Latency, Jitter and Packet Loss

In this section, we outline the evaluation methodology to obtain and analyze network performance metrics within the TSN-DetNet testbed. We use the tool iperf with its 'trip-times' feature to measure one-way latency, jitter, and packet loss. This feature is present in iperf-v2.1.5, built from source code. We also use ping to measure Round-Trip Times. A constant flow of ST traffic is generated with a transmission data rate of $40Mbps$, adhering to the requirements of time-sensitive applications. Concurrent with the ST flow, a BE flow is introduced, with its data rate varied across five distinct rounds to simulate diverse traffic conditions. The BE data rates employed are 40, 60, 100, 200, and $500Mbps$, each sustained for a duration of $15seconds$.

Our Detnet routers are configured to provide dedicated QoS reliability for Scheduled Traffic (ST) flows up to $50Mbps$. Consequently, we limit the ST data rate to comply with this threshold. To analyze the impact of congestion on network metrics, we introduced a $100Mbps$ bottleneck link between the TSN-DetNet gateway and the TSN switch in Chennai. Table I summarizes network performance metrics obtained during peak working hours, capturing the effects of shared network resources on one-way end-to-end latency. We observe that if the total transmission data rate is below 100Mbps, ST and BE flow experiences $6.6ms$ and $6.75ms$ latency respectively. The differences in latency values stem from the fact that ST data is forwarded using label-based forwarding in the core network while the BE flow is forwarded using IP-based forwarding. When the BE data rate exceeds $60Mbps$, pushing the total data rate above $100Mbps$, latencies for both flows increase. However, ST experiences a slight increase in latency and jitter and no packet loss. BE however experiences a $10x$ increase in latency with packet while also suffering from packet loss and increased jitter. We see the same behaviour when the data rate of BE is increased further. The increase in the latency for both flows can be attributed to the queueing of ST and BE packets at the beginning of the congested link.

TABLE I
Analysis of Latency, Jitter and Packet Loss by varying Transmitting Rate

Transmission Rate (Mbps)		Latency (ms)		Loss		Jitter (ms)	
ST	BE	ST	BE	ST	BE	ST	BE
40	40	6.6	6.75	0	0	0.05	0.05
40	60	10	100	0	14%	0.05	0.150
40	100	10	105	0	50%	0.06	0.171
40	500	10	105	0	90%	0.06	0.244

To combat the challenge of an increase in the latency of ST flow and bound the latency of ST flows even with a high BE data rate, we introduce TAPRIO on the TSN switch at the source. We perform measurements during non-peak hours and use the same data rates of ST and BE used in the previous case. We also send ping flows for ST and BE to measure the RTT and observe the effect of TAPRIO on the latency. The TAPRIO's GCL parameters are 450us and 50us of ST and BE respectively. The measurements are presented in Fig 4 and Fig 5. TAPRIO can reduce the effect of the high data rate of BE flow on the latency of ST flow as observed in the last 3 of the 5 sections in both figures. We do see an increase in latency of a few ST packets. We suspect that this could be due to one or two reasons. One of them is TAPRIO being run on the Linux kernel and the kernel not able to schedule TAPRIO at precise intervals due to being involved in uninterruptible tasks. The other reason is the shared links in the MPLS core, which may have other high-priority traffic running on these routers.

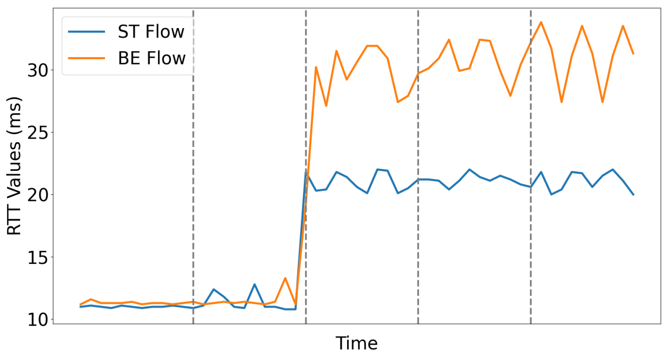

Fig. 4. RTT values for ST and BE flow without TAPRIO

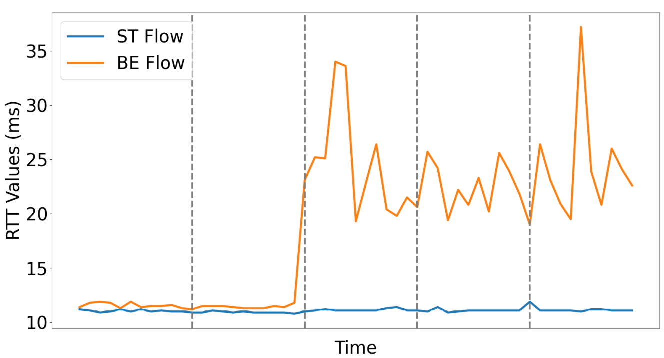

Fig. 5. RTT values for ST and BE flow with TAPRIO

B. PTP Time Synchronization

To assess the synchronizing accuracy between the end-host PTP instances in Bengaluru and Chennai, we run the ptp4l application on both end-hosts, effectively synchronizing their PHCs with each other. The offset measurements were performed concurrently with the latency measurements. The PTP traffic is forwarded using the ST link along with tactile traffic. The follower PTP instance reports synchronization offsets at approximately one-second intervals. Figure 6 visually depicts the synchronization offset observed at the follower PTP instance in relation to the leader PTP instance. This plot was generated using the ptp4l log messages at the follower instance. The root mean square of the synchronization error is $52.48 microseconds$.

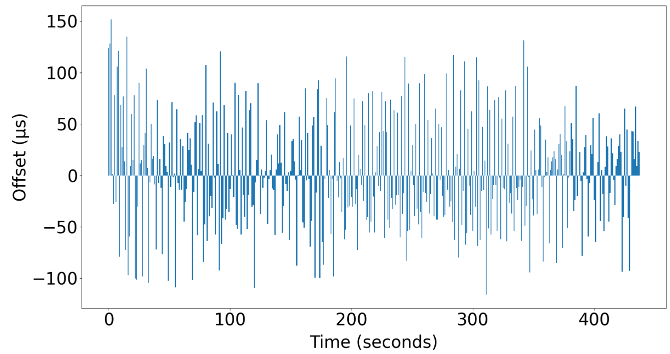

Fig. 6. Synchronization Offset at the follower PTP instance

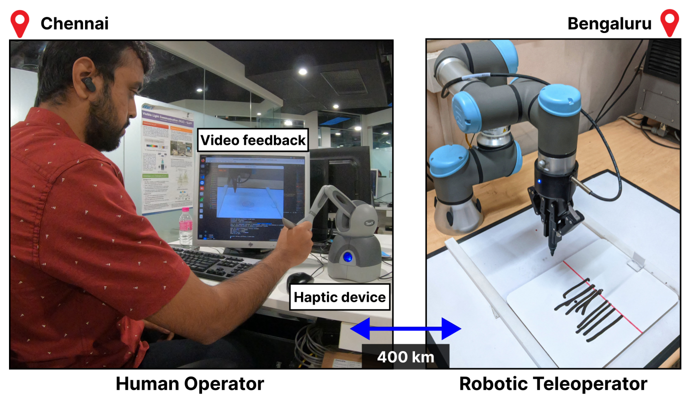

Fig. 7. Teleoperating a robot to draw on a board

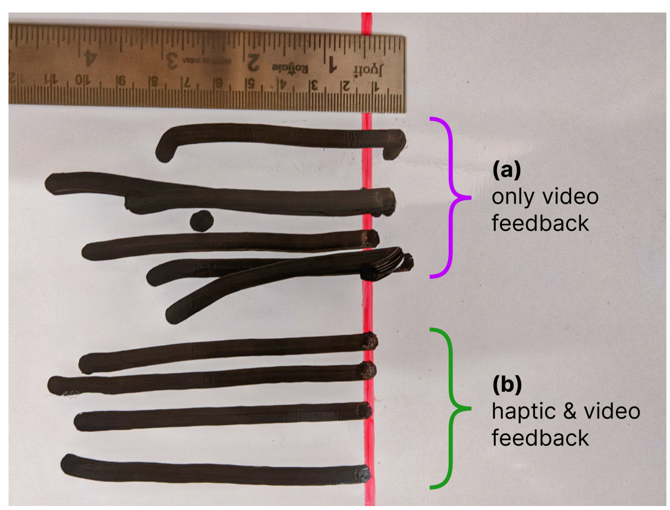

Fig. 8. Drawing with (a) only video feedback; (b) with both haptic and video feedback

C. Robotic Teleoperation

Our testbed facilitates the exploration of various Tactile Cyber-Physical Systems (TCPS) applications, such as robotic teleoperation with feedback mechanisms. We present an experiment analyzing the influence of video and haptic feedback on task performance. The operator remotely traces a horizontal line on a whiteboard using a robot-held marker pen until reaching a pre-defined vertical boundary marked by a red line, as shown in Fig. 7. Initially, the operator relies solely on visual feedback from a 30fps web camera to execute the task. This scenario often leads to inaccurate line tracing, with frequent boundary breaches, as illustrated on the top of Fig. 8. The lack of tactile information results in difficulty judging distances, leading to overshoots. Introducing haptic feedback through the haptic device significantly improves performance and user experience, as illustrated at the bottom of Fig. 8. Force feedback cues are triggered upon surface contact with the whiteboard, providing a sense of ”touch”. Additionally, an alert via force feedback notifies the operator when approaching the boundary, leading to a timely halt and preventing a breach.

In this experiment, the latency between sending a control packet and receiving a video packet is $44.3\ ms$. It comprises of network RTT of $11\ ms$ and a frame duration of $33.3\ ms$. The haptic feedback offers a more instantaneous response to contact and proximity. This real-time feedback loop reduces the cognitive workload of judging the actual robot position from a video feed at a lag. This leads to smoother and more accurate movements. Furthermore, the “touch” sensation enhances the naturalness and intuitiveness of the teleoperation experience, fostering a sense of direct interaction with the environment. It should be noted that the results are subjective in nature, and the performance depends on the time taken to complete the drawing and the experience of the human operator. The average drawing speed by the operator is $0.45\ m/s$ in the experiment in which Fig. 8 was recorded.

VI. Conclusion and Future Work

We presented a TSN-DetNet testbed design for TCPS applications over a WAN link. We implemented TAS and PTP mechanisms using Linux based systems and evaluated the testbed performance in terms of latency, jitter, packet loss, and time synchronization accuracy. We also demonstrated the feasibility of the testbed for a robotic teleoperation application with haptic and video feedback. Our results showed that the testbed can support applications adhering to the latency and reliability requirements stated in 3GPP TS for CPS applications over a WAN link for ST flows. The testbed also achieved microsecond-level time synchronization accuracy, which is essential for realizing TAS and accurate network telemetry measurements.

As part of our future work, we would like to implement in hardware, Frame Replication and Elimination for Reliability [16] and TAS scheduling mechanism on the data plane using programmable smartNICs [17] and test the performance on this testbed. Utilizing all the hardware that is part of the building blocks of the testbed is of prime interest. This testbed can also function as a training ground for real-time teleoperation of robots in applications like telesurgery.

Building a TSN-DetNet testbed for TCPS applications across cities requires collaboration among several organizations and interdisciplinary talent. We hope that the results from our future work on this testbed can help in gaining insights into the feasibility of TCPS applications over TSN-DetNet by the research community members.

Acknowledgements

This work was partly supported by the Ministry of Electronics and Information Technology, Government of India (SP/MITO-20-0006) and partly by the Centre for Networked Intelligence (a Cisco CSR initiative) at the Indian Institute of Science, Bengaluru, India.

References

[1] 3GPP, “Service requirements for cyber-physical control applications in vertical domains; stage 1 (release 19),” *Technical Specification Group Services and System Aspects*, September, 2023.

[2] “Ieee standard for local and metropolitan area networks – bridges and bridged networks - amendment 25: Enhancements for scheduled traffic,” *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)*, pp. 1–57, 2016.

[3] “Ieee standard for local and metropolitan area networks–frame replication and elimination for reliability,” *IEEE Std 802.1CB-2017*, pp. 1–102, 2017.

[4] “Ieee standard for local and metropolitan area networks–timing and synchronization for time-sensitive applications,” *IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011)*, pp. 1–421, 2020.

[5] IETF, “Deterministic networking,” 2023. [Online]. Available: https://datatracker.ietf.org/wg/detnet/about/

[6] K. Polachan, J. Pal, C. Singh, T. Prabhakar, and F. A. Kuipers, “Tcpsbed: A modular testbed for tactile internet-based cyber-physical systems,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, 2021.

[7] N. Arjun, S. Ashwin, K. Polachan, T. Prabhakar, and C. Singh, “An end to end tactile cyber physical system design,” in *2018 4th International Workshop on Emerging Ideas and Trends in the Engineering of Cyber-Physical Systems (EITEC)*. IEEE, 2018.

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

[9] V. Gokhale, K. Kroep, V. S. Rao, J. Verburg, and R. Yechangunja, “Tixt: An extensible testbed for tactile internet communication,” *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 32–37, 2020.

[10] “Touch haptic device,” 3D Systems. [Online]. Available: https://www.3dsystems.com/haptics-devices/touch

[11] “Quest 2,” Meta. [Online]. Available: https://www.meta.com/quest/products/quest-2/tech-specs/#tech-specs

[12] “Cb3,” Universal Robots. [Online]. Available: https://www.universal-robots.com/products/cb3/

[13] “Adaptive grippers,” Robotiq. [Online]. Available: https://robotiq.com/products/2f85-140-adaptive-robot-gripper

[14] “Ft 300-s force torque sensor,” Robotiq. [Online]. Available: https://robotiq.com/products/ft-300-force-torque-sensor

[15] “Depth camera d415,” Intel. [Online]. Available: https://www.intelrealsense.com/depth-camera-d415/

[16] S. K. Rana, H. Verma, J. Pal, D. Choudhary, T. V. Prabhakar, and C. Singh, “Enhancing reliability of scheduled traffic in time-sensitive networks using frame replication and elimination,” in *2023 IEEE 29th International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2023, pp. 1–6.

[17] J. Pal, D. Choudhary, N. K. Gnani, C. Singh, and T. Prabhakar, “μtas: Design and implementation of time aware shaper on smartnics to achieve bounded latency,” *arXiv preprint arXiv:2310.07480*, 2023.