Major Project Report on

Design and Simulation of Haptic Communication over the Tactile Internet in a Time-Sensitive Environment

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

Gaurang Jitendra Velingkar, 191IT113

Rakshita Varadarajan, 191IT140

Stafan Kuttikal Santhosh, 191IT151

under the guidance of

Dr. Geetha V

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575025

April, 2023

DECLARATION

We hereby *declare* that the Report of the Under Graduate Project Work entitled **"Design and Simulation of Haptic Communication over the Tactile Internet in a Time-Sensitive Environment"**, which is being submitted to **National Institute of Technology Karnataka, Surathkal**, in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Information Technology in the Department of Information Technology, is a *bonafide report of the work carried out by us.* The material contained in this Report has not been submitted at any University or Institution for the award of any degree.

Place: NITK, Surathkal
Date: 10/04/2023
.

Gaurang Jitendra Velingkar (191IT113)
Rakshita Varadarajan (191IT140)
Stafan Kuttikal Santhosh (191IT151)

CERTIFICATE

This is to *certify* that the Under Graduate Project Work Report entitled **"Design and Simulation of Haptic Communication over the Tactile Internet in a Time-Sensitive Environment"**, submitted by Gaurang Jitendra Velingkar (191IT113), Rakshita Varadarajan (191IT140), Stafan Kuttikal Santhosh (191IT151) as the record of the work carried out by them, is *accepted as the Under Graduate Project Work report submission* in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Information Technology in the Department of Information Technology.

Dr. Geetha V

Project Guide	Chairman - DUGC and Head
Dept. of Information Technology	Dept. of Information Technology
NITK Surathkal, Mangalore	NITK Surathkal, Mangalore

ABSTRACT

The internet must evolve to suit our needs as we develop into the age of haptic feedback and more complex applications. It has long been a one size fits all service for applications like File Transfer, Real-Time Video, VoIP, Web Traffic, and Transactions. While plenty of work is needed here to suit time-sensitive applications, these applications are at some level functional with the current state of the internet. Haptic Feedback, however, requires ultra-low latency and ultra-high reliability. For applications such as telesurgery, virtual reality, and more, latency conditions for haptic response in the human body must be one millisecond to avoid any noticeable delay. Hence the Tactile internet being developed with these requirements in mind needs to have provisions for time-sensitive networks regarding haptic feedback. We aim to propose strategies or techniques to identify such a delay-sensitive network that is specifically for haptic feedback, ensuring that during the whole process from packet creation to identification and transmission, it takes the least time possible and fulfills the needs of the tactile internet.

Keywords— tactile internet, delay-sensitivity, haptic feedback, configuration

CONTENTS

LIST OF TABLES

Chapter 1

INTRODUCTION

The past decade has seen rapid advancements in technology. Starting with a fixed, text-only internet connection, technology progressed to a mobile, multi-media link, and now towards the Internet of Things, or IoT. The tactile internet has many applications in healthcare, robotics, augmented reality, virtual reality, education, etc. The goal of the tactile internet is to facilitate haptic communication or online communication based on touch. Any technology that can simulate the sensation of touch by exerting forces, vibrations, or motions on the user is called haptic technology (also known as kinaesthetic communication or 3D touch). These technologies enable the creation of virtual objects in computer simulations, their control, and improved remote control of machinery and equipment. The next generation of cyber-physical systems and human-computer interaction are expected to undergo radical changes due to the tactile internet's remarkable capacity for human-like communication.

Time-Sensitive Networking (TSN) ensures Quality of Service (QoS) for time-sensitive applications. To prevent overlap, it uses the guard band mechanism to distinguish between time-sensitive traffic and regular traffic. The various scheduling methods used by TSN are outlined under the IEEE 802.1 standard [4]. While TSN standards mostly talk about scheduling, the problem of finding the best path is left open-ended. This project addresses these issues by developing a mechanism to identify the optimal path for time-sensitive traffic in a Tactile internet network. The time-sensitive traffic considered here is the haptic information. A method to represent haptic communication in the network simulation followed by differentiating haptic traffic from non-haptic traffic has also been proposed.

1.1 Architecture of Tactile Internet

Architectures proposed so far have been application agnostic or application specific. This subsection will focus on application agnostic architectures. Most such architectures aim at enhancing cellular networks to meet the high standards of the tactile internet. We will go through a few of these architectures in brief.

1.1.1 IEEE P1918.1 Architecture

This specific architecture is founded on flexible and modular design concepts. It is made up of network domains and edge domains, as well as functional elements including tactile devices, support engines, and gateway nodes. The suggested architecture supports interoperability, but it doesn't concentrate on the ultra-low latency and ultra-high dependability required for applications that are special to the tactile internet and depend on elements other than the architecture.

1.1.2 MEC-enhanced Cellular Network Architectures

In order to serve Tactile Internet applications, this architecture suggests a multi-level cloud system for offloading in the cellular network. It comprises of micro-cloud, mini-cloud, and core-cloud components, which are the three hierarchical cloud tiers. The mini-clouds and core-cloud units are hierarchical in relation to the micro-clouds, whilst the micro-clouds manage user functions. According to the results, only a small number of users can achieve round-trip latency of 1ms. This study has been expanded upon by other studies.

1.1.3 FiWi-enhanced Cellular Network Architectures

This FiWi-enhanced LTE-A HetNet architecture for the Tactile Internet combines Gigabit WLAN and EPON technologies to offer WiFi offloading and high-capacity fibre backhaul. Both time division multiplexing and wavelength division multiplexing EPONs are taken into account in the EPON fibre backhaul, which utilises an OLT linked by fibre to several ONUs. For stationary users, the design satisfies the ultra-low latency requirements, but not for mobile users.

1.2 Standards for Tactile Internet

The baseline standard aims to define which functionalities and functional entities are to be present in which locations and what the relationships between them would be like.

There are many different tactile applications that may be mapped to the IEEE 1918.1 standard. It is firmly grounded in design tenets like modularity and adaptability. It is composed of edge domains and network domains.

The standard also specifies fundamental language baseline work for the Tactile Internet (TI) so that it may be realised and understood uniformly by manufacturers, operators, end users, and other parties that could use or implement the service or who would otherwise be stakeholders.

Additionally, the standard clearly outlines multiple TI use cases, with the end goal being that a choice between them will be made at the time the TI service is invoked, and the network would be configured accordingly. As a result, use cases are formally defined in IEEE 1918.1.

On top of the baseline, there are extra standards and amendment standards that can be introduced. The working group that developed the standards has included provisions for them. A notable illustration of this is the IEEE 1918.1.1 "Haptic Codecs for the TI" standard. The codecs that are being developed will function on both the baseline standard and applications that are far distant from it, such as over a variety of different networks. This standard has already advanced to a highly advanced point.

1.3 Applications of Tactile Internet

The tactile internet facilitates a wide range of applications from telesurgery, and autonomous driving to virtual reality and more. With each of these use cases, the approach differs. This includes the benefits of just tactile latency such as better synchronization of various smart grid systems. Some of them are as follows.

1.3.1 Healthcare

Exoskeletons are a very interesting field. There have been university experiments where folks confined to wheelchairs have been able to move around using exoskeletons. The bottleneck however happens to be the latency. If we can build wireless systems with tactile latency it opens up tons of opportunities for the disabled.

Tele-surgery is one of the most talked about applications of the tactile internet.

There have been papers that discuss providing wireless connectivity to a master surgical console through which surgeons can remotely manipulate a surgical robot and receive real-time auditory, visual and haptic feedback.

1.3.2 Education and treatment

The Tactile Internet greatly facilitates physiotherapists. With Haptic Feedback one can feel for the injury and Identify the degree of movement from miles away.

There also opens up the possibilities of virtual lab experiences. Interactive sessions are unheard of in the educative industry.

1.3.3 Virtual Reality

Virtual Reality is a much broader definition and extends to a variety of fields. From opening up to interaction at a personal level through transmitting haptic in the metaverse, to opening up a whole new world of products and means of exchanges.

1.4 Motivation

Despite all of its claims and the revolution it is meant to usher in, the tactile internet requires communication through a network with ultra-low latency and ultra-high reliability. It aims for a round trip latency from end to end of less than 1ms. This is appropriate in situations where a robotic arm must respond quickly to haptic communication initiated by a surgeon or in virtual reality settings where a more significant latency might result in cyber-sickness owing to the unsynchronized feedback from several senses [8].

We would need to keep the haptic signal sender and receiver at most 150 km [8] apart to achieve the acceptable latency requirements in an ideal scenario where we anticipate signals to move at the speed of light and have no other kinds of delay. However, in a real-world setting, delays exist because of transmission delays, bandwidth limitations, and queuing delays. As a result, the distance between tactical internet end-points gets even closer as the distance restriction gets less. In light of this, it is crucial to develop methods for lowering network communication lag to support the objectives of the tactical internet.Time sensitive networking combined with the tactile internet is a very viable solution for this very reason, and acts as motivation for this project.

Chapter 2

LITERATURE REVIEW

Tactile Internet aims towards real-time applications with delay constraints. Currently the research is going on in the areas of Haptic Information encoding and decoding, various standards, using existing network to support tactile internet, etc. In this chapter, various works with respects to tactile internet is studied and gaps are identified.

In [8], N. Promwongsa et al. have provided a very comprehensive report on the current state of tactile internet research. They have looked at 75 papers covering all aspects such as Architectures, Protocol and Intelligent Prediction, Radio Resource Allocation Algorithms and Non-radio resource allocation algorithms. With dedicated reading maps depending on the purpose we were able to look at implementations that fullfilled the two main criterias of ultra low latency and ultra high reliability that is characteristic of the tactile internet. We did not find finer details of the various implementations here but an overview of the various approaches and what path we can take. We had to look at papers that were cited here for an in depth overview of the methodologies they have implemented. In [9], D. Rico et al. provide a detailed study about Multi-connection Tactile Internet Protocol (MTIP), implemented on top of Internet Protocol (IP). It is based on a combination of sequence numbers and timestamps in packets, along with a global clock in the network to maintain time. The dispatch algorithm broadcasts packets to the best sub-links. The reception algorithm handles redundancy and re-ordering of packets using sequence numbers and timestamps in the packets. While the paper talks in detail about the algorithms used and protocols implemented, along with formal modeling and verification of the same, improvement in terms of analyzing performance aspects can be done. The protocol improves reliability and reduces latency by selecting best paths to send the packets to, while also successfully solving the problem of dealing with sequence numbers and timestamps to have an ordered flow of data.

In [10], Marshall et al. detail the necessities of a good Quality of Service for a haptic feedback network. It explains the components that make good QoS like delay, jitter, packet loss and throughput. They detail a real end to end implementation all the way from haptic feedback using a Distributed Haptic Virtual Environment. A custom PDF model was then created for use in the network simulation tool OPNET. A simulation model of DHVE applications running over a network was then developed. Moreover the study is particularly relevant to DVHEs that are implemented as networked peers rather than traditional client-server architectures. In [2], D. Van Den Berg et al. provide an outline of the problems in establishing the Tactile Internet, as well as a synopsis of the criteria for haptic communications. Furthermore, potential solutions to these difficulties are presented and debated. The paper determines the most important requirement for enabling haptic communication over the Tactile Internet as an end-to-end delay that is at most 1 ms. It also listed reliability, and security while keeping the end-to-end delay in mind. While the paper theoretically goes in detail, it lacks an implementation backing.

In [4], Norman Finn talks about time sensitive networking (TSN) as a solution for real-time applications that need not only minimal (¡ 1 ms) end-to-end delay, but also for guaranteed upper bounds on end-to-end latency, zero packet loss due to buffer congestion and overall low packet loss because of equipment failure. Overall, the paper discusses TSN in depth, covering essential features, use cases and queuing algorithms of TSN. The paper provides alternatives to TSN and one by one also compare them to TSN, concluding that TSN is the best choice for satisfying the requirements of a real-time application with the previously mentioned prerequisites. [1] by V. Gavriluţ et al. provides an overview of all methods to design networks for time-sensitive applications. It goes through all research being conducted in the field and analyzes the result based on reliability, timeliness and network cost. This work addresses design techniques of distributed systems that ensure timing constraints of applications are met. The paper mentions all challenges related to designing networks for time-critical applications, along with solutions to each of the problems. While it talks about design tasks like network planning, routing, priority and traffic type assignment, bandwidth allocation and scheduling, it does not speak of anything related to its implementation. In [7], A. Alnajim et al. address the issues of scheduling and routing that are not specified in detail in the IEEE TSN Standards. This is done by proposing a Quality of Service (QoS) based path selection algorithm along with various incremental scheduling algorithms and then compares all their performances. The paper's results show that the incremental scheduling algorithms outperformed the QoS based path selection algorithm in terms of both the number of scheduled flows and wasted bandwidth in the process.

Podlesny et al. in [11] try different networking mechanisms for supporting low queuing delay required by time sensitive networks. There are two primary approaches. The first one assumes employing congestion control protocols, the MCP Congestion Control Protocol for the traffic generated by a particular class of applications. Congestion protocols prevent situations where the link bandwidth is exceeded and as a result cause delay. The second approach relies on the router operation only and does not require support from end hosts. This paper does not specifically look at Haptic Feedback but instead looks at all other requirements of various internet applications. The congestion control protocol is focused on use of multiple operation modes, transmission at the constant rate in the stable state, and use of the explicit-communication mechanism for converging to fairness. The paper also presents the RD Network Services for aligning network services with application needs which relies on algorithms at the router end. In [6], Nasrallah et al. opt for a Centralized approach to managing the Network Configuration (CNC) interface for a TSN network to globally manage and configure TSN streams. It has various modules for different functions like admission control and resource reservation. They integrate the CNC in the control plane with Time Aware Shaper (TAS) in the data plane. The CNC approach is the central entity for flow approaches. They also make modifications at the centralized/distriuted level and the various TAS parameters. This paper gives a skeleton to base our work on. They have details on how they went about building the various modules and the logic inside. This approach details the simulation software and also has details on how to identify the TSN specific traffic. While not in great detail it gives us an overview of the steps we need to look at for our methodology.

Subarna Singh in [12] enlists all of the routing algorithms defined under the various IEEE 802.1 standards. The thesis presents the Maximum Scheduled Traffic Load (MSTL) as a metric that can be utilised by any scheduling method to quantify the maximum traffic that is scheduled on a network connection, allowing evaluation of the distribution of data load across the network. It also identifies individual metrics like Flowspan that can be used to measure the quality of a schedule generated by various scheduling algorithms. Weighted algorithms based on the Shortest Path First (SPF) and Equal Cost Multipath (ECMP) incorporating the concept of MSTL have been explored and experimented with. Further, Integer Linear Programming (ILP) based algorithms have been discussed in detail like the Load Aware (LA) and Load and Hop Aware (LHA) algorithms. The author also introduces a generic algorithm based on heuristic algorithm Tabu Search aimed at optimizing the MSTL value. The LA and LHA algorithms are seen to perform much better than the weighted algorithms which in turn perform better than SPF and ECMP. Similarly, K. Huang et al. in [13] introduce the concept of scheduling periods in routing algorithms to generate traffic schedules with minimal collisions when their combinability is high.

In [8], N. Promwongsa et al. have provided a very comprehensive report on the current state of tactile internet research. They have looked at 75 papers covering all aspects such as Architectures, Protocol and Intelligent Prediction, Radio Resource Allocation Algorithms and Non-radio resource allocation algorithms. With dedicated reading maps depending on the purpose we were able to look at implementations that fullfilled the two main criterias of ultra low latency and ultra high reliability that is characteristic of the tactile internet. We did not find finer details of the various implementations here but an overview of the various approaches and what path we can take. We had to look at papers that were cited here for an in depth overview of the methodologies they have implemented. In [9], D. Rico et al. provide a detailed study about Multi-connection Tactile Internet Protocol (MTIP), implemented on top of Internet Protocol (IP). It is based on a combination of sequence numbers and timestamps in packets, along with a global clock in the network to maintain time. The dispatch algorithm broadcasts packets to the best sub-links. The reception algorithm handles redundancy and re-ordering of packets using sequence numbers and timestamps in the packets. While the paper talks in detail about the algorithms used and protocols implemented, along with formal modeling and verification of the same, improvement in terms of analyzing performance aspects can be done. The protocol improves reliability and reduces latency by selecting best paths to send the packets to, while also successfully solving the problem of dealing with sequence numbers and timestamps to have an ordered flow of data.

An application layer protocol, called HoIP, or Haptics over Internet Protocol has been introduced in [15], using UDP in the transport layer. The authors have used a haptic signal generator to generate haptic signals and sampled it using two different adaptive samplers, the Weber sampler and the level crossings sampler. The paper compares results of the two methods, both of which ultimately reduce the packet transmission rate. The contributions of the paper have been used as a standard for research related to haptic communication over the internet.

A summary of literature survey can be found in Table 2.0.1 below.

Author	**Methodology**	**Observation**
V. Gavriluţ et al. [1]	The authors provide an overview of all methods to design networks for time-sensitive applications and analyze the result on the basis of delay, throughput and other parameters to ensure that timing constraints are met.	They discuss all challenges related to designing networks for time-critical applications. While they talk about design tasks like network planning, routing, priority and traffic type assignment, bandwidth allocation and scheduling, it does not speak of anything related to its implementation.
D. Van Den Berg et al. [2]	The authors provide details of the problems along with solutions in establishing the Tactile Internet, as well as a synopsis of the criteria for haptic communications.	They determine the most important requirement for enabling haptic communication over the Tactile Internet as an end-to-end delay that is at most 1 ms. They also add to this list reliability, and security while keeping the end-to-end delay in mind. While the paper theoretically goes in detail, it lacks an implementation backing.

Z. Cao et al. [3]	The authors use the SG-WRR strategy to reduce average scheduling intervals. BE Traffic queues are given scheduling opportunities in presence of AVB and TT traffic. Guard band waste is reduced using Dynamic Programming algorithms.	Under their experimental conditions, which have been overly simplified, the bandwidth utilisation increases by 5.66% and the scheduling chances for BE traffic rise by an average of 21.9%.
Norman Finn [4]	The author talks about time sensitive networking (TSN) in depth, covering essential features, use cases and queuing algorithms of TSN.	The author provides alternatives to TSN (such as weighted fair queuing, prioritization, congestion detection) and also compare them to TSN, concluding that TSN is the best choice for satisfying the requirements of a real-time application with low latency.

A. B. D. Kinabo et al. [5]	The authors try to implement Time Sensitive Networking (TSN) over the 802.11ac Wi-Fi standard and analyse the results comparing it to that of wired TSN, while studying the factors obstructing the way of wireless TSN and emulates TSN in Wi-Fi.	The obtained results showed that while TSN is supported over a interference-free Wi-Fi channel, the efficiency of this is very limited (due to lack of reliability thanks to collision and more) and Further improvements have to be made to WiFi's mode of operation to make it more specific for time-sensitive applications. More features should be included into the Wi-Fi standard to include support for TSN specifically in terms of priority schemes.

Nasrallah et al. [6]	They opt for a Centralized approach to managing the Network Configuration (CNC) interface for a TSN network to globally manage and configure TSN streams. It has various modules for different functions like admission control and resource reservation. They integrate the CNC in the control plane with Time Aware Shaper (TAS) in the data plane. The CNC approach is the central entity for flow approaches. They also make modifications at the centralized/distributed level and the various TAS parameters.	This paper gives a skeleton to base our work on. They have details on how they went about building the various modules and the logic inside. This approach details the simulation software and also has details on how to identify the TSN specific traffic. While not in great detail it gives us an overview of the steps we need to look at for our methodology.

A. Alnajim et al. [7]	The authors address the issues of scheduling and routing that are not specified in detail in the IEEE TSN Standards. This is done by proposing a Quality of Service (QoS) based path selection algorithm along with various incremental scheduling algorithms and then compares all their performances.	Contrary to expected, the results show that the incremental scheduling algorithms outperformed the QoS based path selection algorithm in terms of both the number of scheduled flows and wasted bandwidth in the process.

N. Promwongsa et al. [8]	This is a very comprehensive report on the current state of tactile internet research. They have looked at 75 papers covering all aspects such as Architectures, Protocol and Intelligent Prediction, Radio Resource Allocation Algorithms and Non-radio resource allocation algorithms. With dedicated reading maps depending on the purpose we were able to look at implementations that fullfilled the two main criterias of ultra low latency and ultra high reliability that is characteristic of the tactile internet.	The paper is a great introduction to the tactile internet and is a very comprehensive overview of its scope. We did not find finer details of the various implementations here but an overview of the various approaches and what path we can take. We had to look at papers that were cited here for an in depth overview of the methodologies they have implemented.
D. Rico et al. [9]	The authors provide a detailed study about Multi-connection Tactile Internet Protocol (MTIP). Packets are broadcasted to the best sub-links. Redundancy and re-ordering of packets on receiving is handled using sequence numbers and timestamps from the packets.	The protocol improves reliability and reduces latency by selecting best paths to send the packets to, while also using sequence numbers and timestamps to have an ordered flow of data. Further analysis of performance aspects can be done.

Podlesny et al. [11]	The authors have tried different networking mechanisms for supporting low queuing delay required by time sensitive networks by employing congestion control protocols, like the MCP Congestion Control Protocol for the traffic generated by a particular class of application, and by relying on the router operation only.	The authors have not explored Haptic Feedback but instead looked at all other requirements of various internet applications. The congestion control protocol is focused on use of multiple operation modes, transmission at the constant rate in the stable state, and use of the explicit-communication mechanism for converging to fairness.
Subarna Singh [12]	This is an elaborate thesis on different routing algorithms defined under the IEEE 802.1 standard. The author also proposes some improvements over these incorporating MSTL and introduces a generic algorithm based on heuristic algorithm Tabu Search aimed at optimizing the MSTL value.	The weighted Shortest Path and Equal Cost Multipath algorithms show considerable improvements over their unweighted counterparts. The Load and Hop Aware algorithm, however, performs the best in the lot.

K. Huang et al. [13]	The authors introduce the concept of scheduled periods of flows in routing algorithms to load balance traffic across different flows, with an aim to share bandwidth without conflicts, thus overcoming the scheduling bottleneck.	A novel metric to analyze traffic based on combinability of traffic flows was introduced. The proposed method worked better as compared to the shortest path and load balanced routing when the combinability of flows is high, but did not show any significant improvement when the combinability of flows is low.
Falk et al. [14]	The authors introduce NesTING, a network simulator based upon OMNet++/INET framework for IEEE 802.1 TSN standards. Our contributions include time-aware and credit-based shaping simulation model components. Along with VLAN tagging, NeSTiNg offers a framework for switchlocal clocks that may be used to simulate switches without precisely synchronised clocks. These features facilitate strict-priority scheduling as a result.	While the proposed framework is open-sourced and thorough, it still lacks implementations of some standards.

Table 2.0.1: Summary of Literature Survey

2.1 Outcome of Literature Review

Following are the challenges identifies based on the literature review:

- Time Sensitive Networking in Tactile Internet need further implementation and development.
- Time Sensitive Networking on Wireless Networks (WiFi) needs further research.
- Haptic communication over a network by devising a method to encode haptic information and simulate haptic communication over a network based on various traffic conditions in existing networks needs further research.
- Further development of existing standards for communication need to be enhanced to support the tactile internet.

2.2 Problem Statement

As per the inferences drawn from the literature survey, we have formulated the following problem statement:

The design and development of haptic communication and routing for the Tactile Internet.

2.3 Objectives

From the problem statement, the following objectives can be derived for this project:

- Devise a mechanism to represent haptic information for communication over the network.
- The design and development of an algorithm to prioritize haptic communication in the network.

- The design and development of a routing algorithm to support Tactile Internet in the existing network.

Chapter 3

METHODOLOGY

A brief overview of the proposed methodology has been given, with a strong focus on the simulation of haptic communication.

3.1 IEEE 802.1 Standards for Time Sensitive Networking

The IEEE 802.1 TSN standards encompass three fundamental components necessary for building a comprehensive real-time communication solution utilizing switched Ethernet networks that provide deterministic quality of service (QoS) for point-to-point connections. Each standard specification is self-contained and can be utilized independently. However, to unlock the full capabilities of TSN as a communication system, it must be employed in a coordinated manner.[17] The three fundamental components are as follows:

1. **Time Synchronization**
   To ensure real-time communication, it is important for all devices involved to have a shared understanding of time. To achieve this, the IEEE 802.1AS Timing and Synchronization for Time-Sensitive Applications standard is utilized.

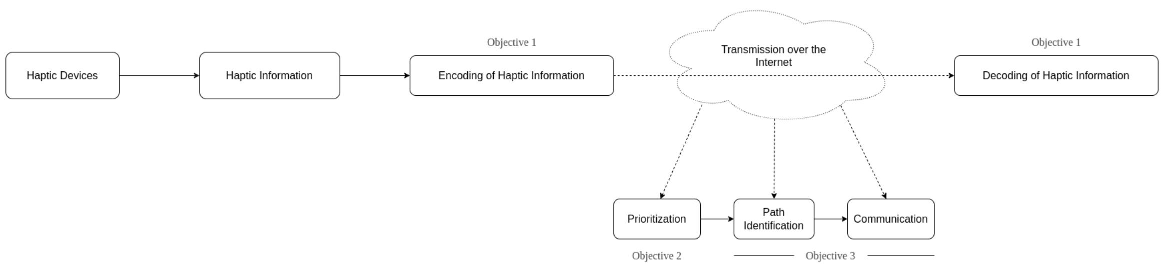
2. **Scheduling and Traffic Shaping**
   To ensure smooth processing and forwarding of communication packets, all devices involved in real-time communication adhere to the same regulations. Traffic Shaping is a method of uniformly distributing packets over time to maintain traffic balance. A couple of standards are defined for this including IEEE 802.1Q Strict Priority Scheduler, IEEE 802.1Qav credit-based traffic shaper, IEEE 802.1Qbv Enhancements to Traffic Scheduling: Time-Aware Shaper (TAS).

3. **Communication path selection, path reservations, and fault tolerance**
   When it comes to selecting communication paths and reserving bandwidth and

time slots in real-time communication, all devices follow uniform regulations. Additionally, these devices may use multiple concurrent paths to ensure fault tolerance. For this, the IEEE 802.1Qca Path Control and Reservation (PCR) standard is used.

3.2 Proposed System

Figure 3.2.1: High-Level System Architecture



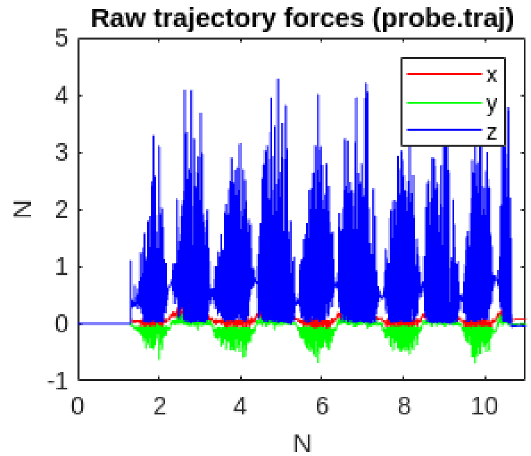
End-to-end latency must be between 1ms to 10ms for the Tactile Internet. Communication must also be dependable since it is used in safety-critical systems in the sectors of healthcare, robotics, education, and other things. The suggested system may be divided into multiple stages in order to implement haptic communication via the internet and promote further study in the area.

3.2.1 Encoding and Decoding of Haptic Information

To send across the network, haptic information at the application layer needs to be encoded into packets. The creation of a system for classifying and ordering haptic packets is necessary. To proceed in the desired direction, the following stages have been identified:

- Identify suitable packet size of packets to be sent over the network without affecting latency
- Identify the type of information that has to be stored in the packets and other constraints
- Ensure that the encoding and decoding process does not adversely affect latency

Figure 3.2.2: Haptic Signal for forces applied by a plane surface



We've used haptic data from the Data Repository for Haptic Algorithm Evaluation. Fig. 3.2.2 shows a sample signal from the dataset. There are two datasets in the Data Repository, one is the force trajectory for a plane and the other for a duck. The dataset is a collection of position and force coordinates in all directions, of the haptic signal sampled at a rate of 500Hz.

Packet rates for modern internet systems, such as VoIP systems, high speed video and audio streaming platforms and live gaming systems, lie within the 100 packets/s range. In comparison, a packet rate of 1000 packets/s is significantly high, and while it's manageable without affecting latency with high-speed internet connections, it can cause a bottle-neck in networks with a limited bandwidth. For this reason, it is necessary to reduce this packet rate, without affecting the quality of the application.

We've used Adaptive Sampling to reduce the packet rate for haptic packets. Adaptive sampling generates packets only when there are considerable changes from previous packets, while maintaining a minimum packet rate. We've considered two adaptive samplers - the Weber Sampler and the Level Crossings Sampler.

- *Weber Sampler* considers the next packet to be generated, at time t ($P_n(t)$) based on the last generated packet (P_{n-1}), such that,

$$|\frac{P_n(t) - P_{n-1}}{P_{n-1}}| \geq \delta$$

where δ is the Weber constant.

- *Level Crossings Sampler* considers the next packet to be generated, at time t ($P_n(t)$) based on the last generated packet (P_{n-1}), such that,

$$|P_n(t) - P_{n-1}| \geq c$$

where c is the Level Crossings threshold.

These constants are significantly dependent on the user and the haptic device. These constants also significantly impact the packet rate. A high threshold will have a very low packet rate. Sometimes, if the haptic signal changes slowly, or if the threshold is high, packet rate would be low. This calls for a minimum transmission rate to have a reliable connection between the sender and receiver. We've considered a minimum rate of 5Hz for our experiments. This can be varied based on the application.

We use UDP to send packets across, since we want minimal delay in transmission. This, however, comes at the cost of reliability. There is also the case of a low packet transmission rate when the haptic signal is constant. This calls for a need to predict the change of the haptic signal at the receiver, when packets aren't received for long periods of time. Hence, we use extrapolation at the receiver, using one of Weber constant or Level Crossings threshold, depending on the adaptive sampler used for sampling at the sender. This adds to processing delay at the receiver's end while sending the haptic response. Some of the extrapolation methods that can be used have been mentioned below.

- *Sample-and-hold*: The signal rendered at time t_n is the same as that rendered at t_{n-1}. This has a much lesser processing delay than other methods.

- *Linear*: This considers the two latest packets received to extrapolate data till a new packet is received. It ensures that when passed through an adaptive sampler, the same samples are obtained as on the true signal. For linear crossings, define $\underline{P}_{n-1} = P_{n-1} - c$ and $\overline{P}_{n-1} = P_{n-1} + c$. For Weber's sampler, the same can be defined as $\underline{P}_{n-1} = (1 - \delta)P_{n-1}$ and $\overline{P}_{n-1} = (1 + \delta)P_{n-1}$. The limiter function, $F(a, b; x)$, can be defined as,

$$F(a, b; x) = \begin{cases} a & \text{if } x < 0 \\ x & \text{if } a \leq x \leq b \\ b & \text{if } x > b \end{cases}$$

Here,
a $= \underline{P}_{n-1}$,
$b = \overline{P}_{n-1}$,
$x = P_{n-1} + \frac{(t - t_{n-1})}{(t_{n-1} - t_{n-2})}(P_{n-1} - P_{n-2})$

Figure 3.2.3: Low-Level System Architecture

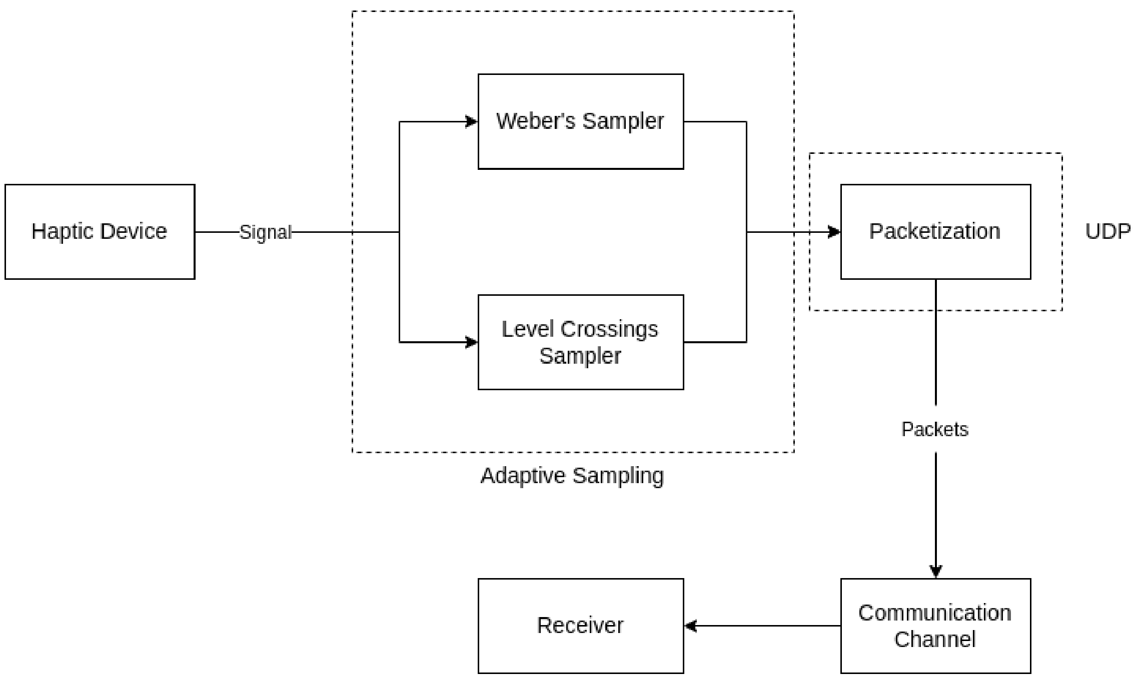


Fig. 3.2.3 depicts how a haptic device generates a signal, generally sampled at 1kHz. This can be improved using one form of adaptive sampling. With a much lower packet rate, a UDP header is added to all of the packets and sent over the communication channel to the receiver.

3.2.2 Packet Structure

The packet structure in the application layer being followed will be similar to the proposed HoIP packet structure [15] with minor adjustments. The proposed frame structure has a header of 16 bytes (3 bytes reserved for future enhancements) followed by the haptic payload. This frame structure can be seen in Figure 3.2.4. The various field are described along with their sample values in Table 3.2.1.

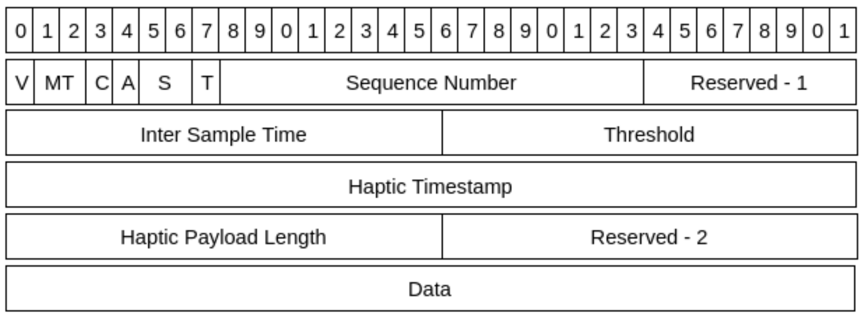


Figure 3.2.4: Proposed HoIP Frame Structure for only Haptic Data

Table 3.2.1: Sample fields in the Packet Structure

Field	**Description**	**Bits**	**Specific Allowed Values (if any)**
V	HoIP Protocol Version Number	1	0
Type	Media Type - 0 for normal data, 1 for Haptics, 2 for Haptics-Audio, 3 for Haptics-Video	2	00 - Normal Data, 01 - Only Haptics, 10 - Haptics-Audio, 11 - Haptocs-Video
C	Type of Content transmitted as payload	1	0 for force coordinates, 1 for position and velocity coordinates
A	Adaptive sampling applied or not	1	0 if not, 1 if it is applied

S	Type of adaptive sampling involved, 1 bit is reserved for future enhancements	2	0 in the case of Weber Sampling, 1 in the case Level Crossings Sampling
T	If threshold is set or not	1	0 if not, 1 if it is applied
Sequence Number	To ensure no packet loss and maintain order	16	-
Reserved - 1	Reserved for future enhancement	8	-
Inter Sample Time	packet delays can be calculated using this, its the gap between inter sample times	16	-
Threshold	threshold value in percentage	16	-
Haptic Timestamp	Timestamp associated to haptic data packets	32	-
Length of payload	length of payload	16	-
Reserved - 2	Reserved for future enhancement	8	-
Data	Sample data as mentioned in Table 4.1.1	64-72 bytes	-

This HoIp is an application header. In the Transport Layer, UDP is the considered protocol and thus UDP header is added. UDP is the simplest transport layer protocol, designed to send data packets. The UDP header is of 8 bytes and has 4 components. The source address, the destination address, the length and the checksum. The source and destination address of the packet changes according to the topology. For haptic packets in most of our topologies, it does not change. However, if there are multiple controllers it will change.

Finally, in the IP layer, IP header of 20 bytes is added. All of these headers combined with the data make the final packet having 116 bytes if only haptic data is considered.

3.2.3 Transmission over the Internet using HoIP

Any type of traffic may go through the internet, including text, audio, and video. In addition to haptic information, communication channels enable the transfer of all common types of network packets. Time-Sensitive Networks prioritize audio-video packets to provide smooth streaming services to users on the network. Considering haptic communication also happens in a time-sensitive environment, it is important to give it the highest priority.

This haptic communication can happen in two methods - one being by Three Separate Streams and the other by Augmenting Data along with Haptic Data.

Transmission using Three Separate Streams

The first method has three separate streams over the transmission medium (we consider the three separate streams as one having haptic data, other carrying the associated video data and the last one carrying associated audio data). Each of these streams are assigned a priority, the highest being that of haptic data as it needs to be sent the earliest, followed by the other two. This priority can be set with the help of PCP tags, and the scheduling and traffic shaping can be done using any of the TSN Traffic Shaping Standards, such as the IEEE 802.1Qav credit-based traffic shaper.

Transmission by augmenting Haptic Data along with Media Data

Alternatively, the second method for haptic communication consisting of haptic, audio and video datacan be done with the help of a transmission algorithm which is capable of prioritizing the appropriate media frame (audio or video), and augmenting (adding/combining) with the immediate haptic data present. This can be wrapped into a packet and be sent across the transmission medium. This augmentation of media multiplexing enables transmission of these media frames on a priority basis.

While data other than haptic (audio/video data) can go in separate streams as their own data, this method of haptic communication allows complete utilization of maximum ethernet frame size (1500 bytes for payload) by sending the data along with haptic data.

The detailed methodology begins with three buffers - haptic data buffer, audio data buffer and video data buffer. Packets to be transmitted through the medium are put into the buffers depending upon the type of data. Prior to transmission across the network, augmentation of the haptic data along with the audio/video data takes place. According to this augmentation, the appropriate header associated to the type of augmentation is appended to the payload in the application layer. Following this in the IP layer, the IPv4 header of 20 bytes is added along with the UDP header of 8 bytes. In order to identify which type of augmentation is taking place, 2 bits are used from the Type Of Service field, particularly Bit 6 and 7 from the 8 Bits in ToS field) in the IPv4 header. These two bits are currently reserved for future use, and thus we use them for oOn varying these 2 bits, the different augmentations are represented.

The payload value allowed for the augmented data (audio/video frame value) is decided by subtracting the sum of the packet header value (H bytes) and the haptic data payload value (HapP bytes) from the maximum ethernet transmission frame size (1500 bytes), i.e, 1500 - (H + HapP) bytes.

There are four scenarios of augmentations possible as listed below -

1. **Normal Data Being Sent (Non Haptic, Audio or Video Data)**:
   If all three buffers are empty, normal data is sent across the transmission medium as usual. The ToS 6th and 7th bits in the IPv4 header are set to 0 each to indicate this type of augmentation. The packet structure for this is shown in Figure 3.2.5.

Figure 3.2.5: Packet Structure in the case 1 - Normal Data Being Sent

2. **Haptic Data only**:
   In the case only the haptic buffer having packets, the haptic data is sent as is. The ToS 6th and 7th bits in the IPv4 header are set to 0 and 1 respectively to indicate this type of augmentation. Further, this packet has the application

header of 16 bytes and associated payload for haptic data between 64-72 bytes (Haptic Payload - HapP) as proposed in the previous section. This packet can be seen in Figure 3.2.6, while the specific HoIP Header can be visualized as shown in the previous section in Figure 3.2.4.

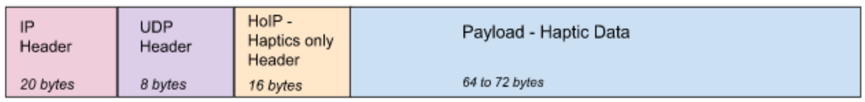


Figure 3.2.6: Packet Structure in the case 2 - Haptic Data only

3. **Haptic Data along with Audio Data only**:
   In the case the haptic buffer and audio buffer is having packets, the haptic data is augmented with the audio data into a packet as shown in Figure 3.2.7, and this is transmitted through the network. This packet also an added HoIP Haptics + Audio Application header of 24 bytes which is depicted Figure 3.2.9 (fields added by Video can be see in the Table XX). The ToS 6th and 7th bits in the IPv4 header are set to 1 and 0 respectively to indicate this type of augmentation. The associated payload value available for audio data is taken as the remaining value of bytes available after applying the constraint of maximum ethernet transmission value (1500 - (H + HapP) bytes). In this case, the audio data payload is restricted between 1,376 to 1,384 number of bytes.

Figure 3.2.7: Packet Structure in the case 3 - Haptic Data along with Audio Data only

4. **Haptic Data along with Video Data**:
   In the case the haptic buffer and video buffer is having packets, the haptic data is augmented with the video data into a packet as shown in Figure 3.2.8, and this is transmitted through the network. This packet also an added HoIP

Haptics + Video Application header of 24 bytes which is depicted in Figure 3.2.9 (fields added by Video can be see in the Table XX). The ToS 6th and 7th bits in the IPv4 header are set to 1 and 1 respectively to indicate this type of augmentation. The associated payload value available for video data is taken as the remaining value of bytes available after applying the constraint of maximum ethernet transmission value (1500 - (H + HapP) bytes). In this case, the video data payload is restricted to a maximum of 1,376 to 1,384 number of bytes.

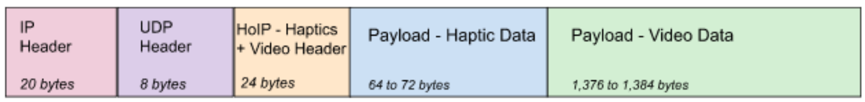


Figure 3.2.8: Packet Structure in the case 4 - Haptic Data along with Video Data

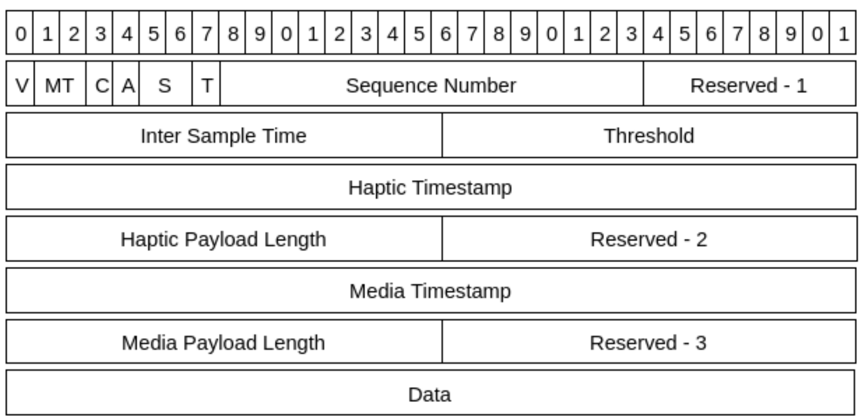


Figure 3.2.9: Proposed HoIP Frame Structure for Haptic + Audio/Video Data

In the case of data being present in haptic, audio and video buffer at the same time, multiplexing happens which ensures that the frames of the media type having priority (audio or video) are augmented first with haptic data and sent across. The detailed algorithm as referred from [16] can be found in Algorithm 1 below.

3.2.4 Routing Algorithm

We've used the Resource Reservation Protocol (RSVP) as a signalling protocol to reserve bandwidth for communication and the Shortest Path algorithm based on path

Algorithm 1 Transmission Algorithm

```
if both audio & video frames are not there then
   send the immediate haptic data;
   instructions2;
else if only video frames are there then
   while immediate video frame is not fully transmitted do
      augment the immediate video frame with immediate haptic data;
      send the augmented data;
   end while
else if only audio frames are there then
   while immediate video frame is not fully transmitted do
      augment the immediate audio frame with immediate haptic data;
      send the augmented data;
   end while
else
   while prioritized media frame is not fully transmitted do
      augment the immediate media frame with immediate haptic data;
      send the augmented data;
   end while
end if=0
```

delay to route traffic in the network. In computer networks, RSVP is a signalling mechanism intended to allocate resources along a path for a certain data flow. In order to provide Quality of Service (QoS) for applications like multimedia streaming, video conferencing, and real-time communication, RSVP runs at the transport layer. In order to guarantee that the data flow obtains the appropriate QoS, each network device along the dedicated path between the source and destination hosts reserves the necessary resources, such as bandwidth and buffer space. Due to RSVP's usage of a soft state approach, resources are released when they are no longer required and the reservation state is frequently refreshed. RSVP can operate in both unicast and multicast environments, and it can coexist with other protocols, such as TCP and IP, to provide end-to-end QoS guarantees.

As mentioned previously, RSVP uses a soft-state mechanism, which means they maintain the state of resource reservations using periodic refresh messages. It has two types of messages: *Path* messages and *Resv* messages. *Path* messages are used to signal the desired path and required resources for the data flow, while *Resv* messages are used to confirm the reservation of resources along the path.

The procedure followed by the RSVP algorithm involves the following steps:

- The sender sends a *Path* message to the destination host, specifying the required resources and the path.
- Each router along the path receives the *Path* message and reserves the necessary resources.
- The destination host sends a *Resv* message back to the sender, confirming the reservation.

The basic RSVP algorithm reserves path based on available bandwidth. However, our application requires a steady connection latency of less than 10ms. As such, a minor modification to the RSVP protocol is required in order to reserve resources only when the entire communication latency in the path is below a certain threshold. This modification also requires evaluating whether there is enough available bandwidth to facilitate communication. These evaluation checks are performed at regular intervals and the routing table of the router is updated accordingly. If resources are to be reserved before the start of the communication between the source and destination, the RSVP algorithm must find the best path and reserve bandwidth as per need before start of the communication. Packet delay in each of the routes from the source must be taken into account to determine the optimum path. The packet delay of the *Path* and *Resv* messages used to reserve bandwidth for communication may not be reliable since the delay can vary once communication starts. To increase the accuracy of the algorithm, it is crucial to take delay into account over a long period of time and choose the path where delay fluctuates the least.

A table of history of communication delay over a path is therefore required to be maintained. This can help select the best reliable path suitable for communication. The RSVP and Shortest Path algorithms should be modified accordingly to incorporate this property. The modified shortest path algorithm is described in Algorithm 2 below.

β in the above equation is a constant that determines the weight to be given to variance of delay in an edge. The modified RSVP algorithm is described in 3 below. γ is the threshold for maximum allowed delay between source and destination, which in our case is 10ms. α is the maximum allowed variance of delays in the path. This

Algorithm 2 Shortest Path Algorithm

```
edges ← list of edges in network
history ← table of delays throughout the day in each edge
for edge in edges do
    src ← edge.src
    dest ← edge.dest
    edge.weight ← history[edge].mean + β * history[edge].std
    if src.total_delay + edge.weight ≤ dest.total_delay then
        dest.total_delay ← src.total_delay + edge.weight
    end if
end for=0
```

ensures that whatever path is chosen is steady. Variance in each edge and path is represented using standard deviation.

Algorithm 3 Modified RSVP Algorithm

```
paths ← ordered list of paths from source to destination
history ← table of delays throughout the day in each path
for path in paths do
    if history[path].mean ≥ γ then
        disable haptic communication
    else if history[path].mean + history[path].std ≥ γ then
        disable haptic communication
    else if history[path].std ≥ α then
        disable haptic communication
    else
        allow haptic communication
    end if
end for=0
```

The two algorithms depend heavily on history being stored in history tables. If the tables aren't initialised with the required values, they need to be added to the tables as and when obtained. Values in the tables are refreshed with newer, latest values when available. This ensures that the tables are always up to date with the latest data.

Chapter 4

RESULTS AND ANALYSIS

The primary objective of our experiments is to test current TSN standards to see if they enable haptic communication and to identify the applications that would be most suited for certain networks, whether they are based on entertainment or more critical applications like healthcare and self-driving, etc. Various improvements to routing algorithms, scheduling and time synchronization can be tried out and their overall impact on the traffic flow within the network can be analysed.

4.1 Haptic Simulation and Encoding

4.1.1 Dataset

We've used haptic data from the Data Repository for Haptic Algorithm Evaluation. There are two datasets in the Data Repository, one is the force trajectory for a plane and the other for a duck. Each of these datasets contain four types of data -

1. Raw Trajectory - The subject is scanned and a model of its surface is obtained. A force sensor is then manually used to scan the body of the object and collect position and force related data.

2. Out Trajectory - This is a slightly modified version of the raw trajectory to normalise points when the sensor hasn't been in contact of the object surface or when there has been noise.

3. In Trajectory - This is a hypothetical trajectory of forces applied from within the object surface to obtain the out trajectory. Since sensors cannot penetrate into the object body, this is analytically computed.

4. Haptic rendering - An algorithm generates a stream of forces when the in trajectory is fed to it. This resembles forces applied to a haptic device by the object.

Each of these data types consist of seven attributes, which are x, y and z coordinates of the position of a point of the object, forces applied in all three directions and a timestamp. The signal has been sampled at 500Hz, i.e., at an interval of 0.002s. Table 4.1.1 lists some sample values from the dataset.

Attribute	**Value**
t	5.9820
x	-2.815819
y	-6.819908
z	22.773463
$Force_x$	0.084810
$Force_y$	0.005177
$Force_z$	0.054419

Table 4.1.1: Sample data from dataset

4.1.2 Adaptive Sampling

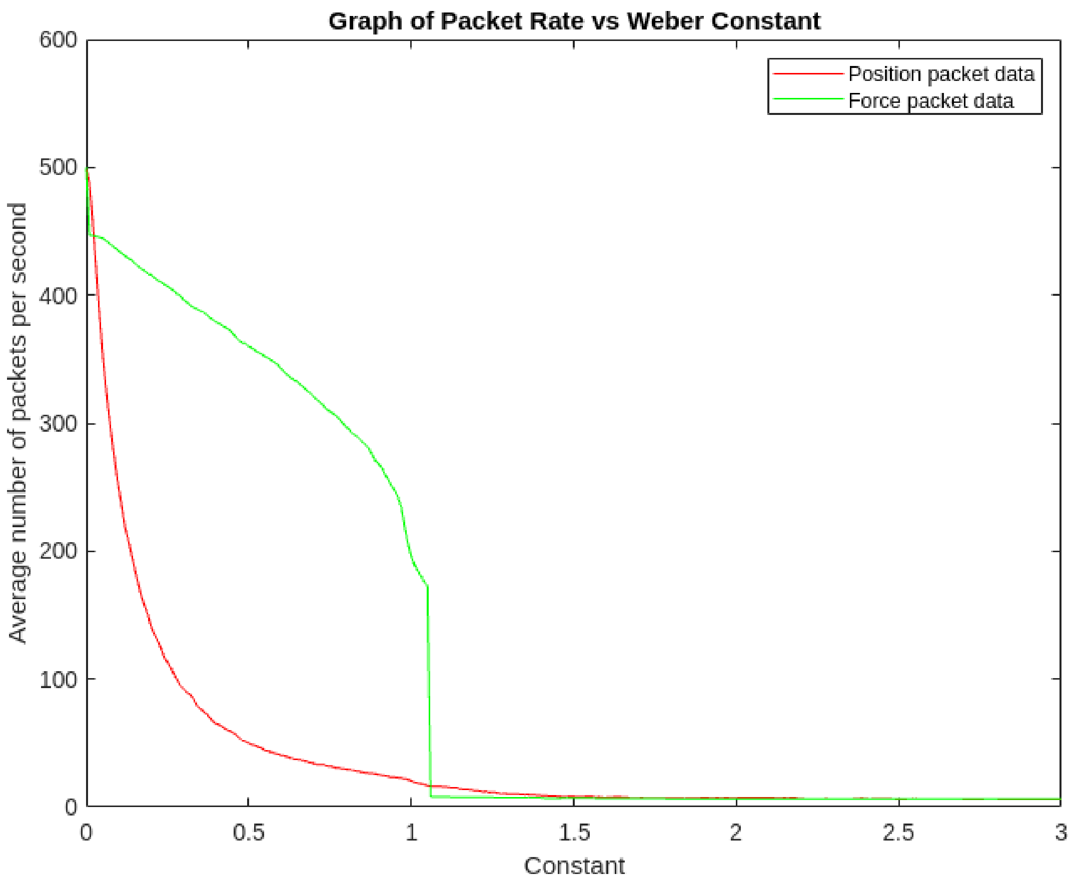
In our experiments for adaptive sampling, we've considered different threshold values of each sampler for position coordinates and force coordinates. We've conducted various experiments to analyze the results.

Fig. 4.1.1 shows that packet rate reduces exponentially as the Weber constant changes. This implies that position of the object is constantly changing, and hence any change in threshold at any point drastically reduces the packet rate. The same cannot be said about the force data, as the change is relatively much slower for lower values of threshold but there is a steep slope as the Weber Constant crosses 1.

On the other hand, Fig. 4.1.2 shows that packet rate reduces exponentially as the threshold changes, similar to the Weber Threshold graph, but the change in the forces data is much more drastic. In such a case, it is harder to identify a good value for the threshold, because of how drastic the change is.

If we are to encode position and force related data into one packet, it is intuitive to use the threshold for packet data with a higher packet rate. For Weber's Sampling, that can be the threshold considered for force related data, before the graph drastically drops. This is because packet data can be linearly extrapolated without any change

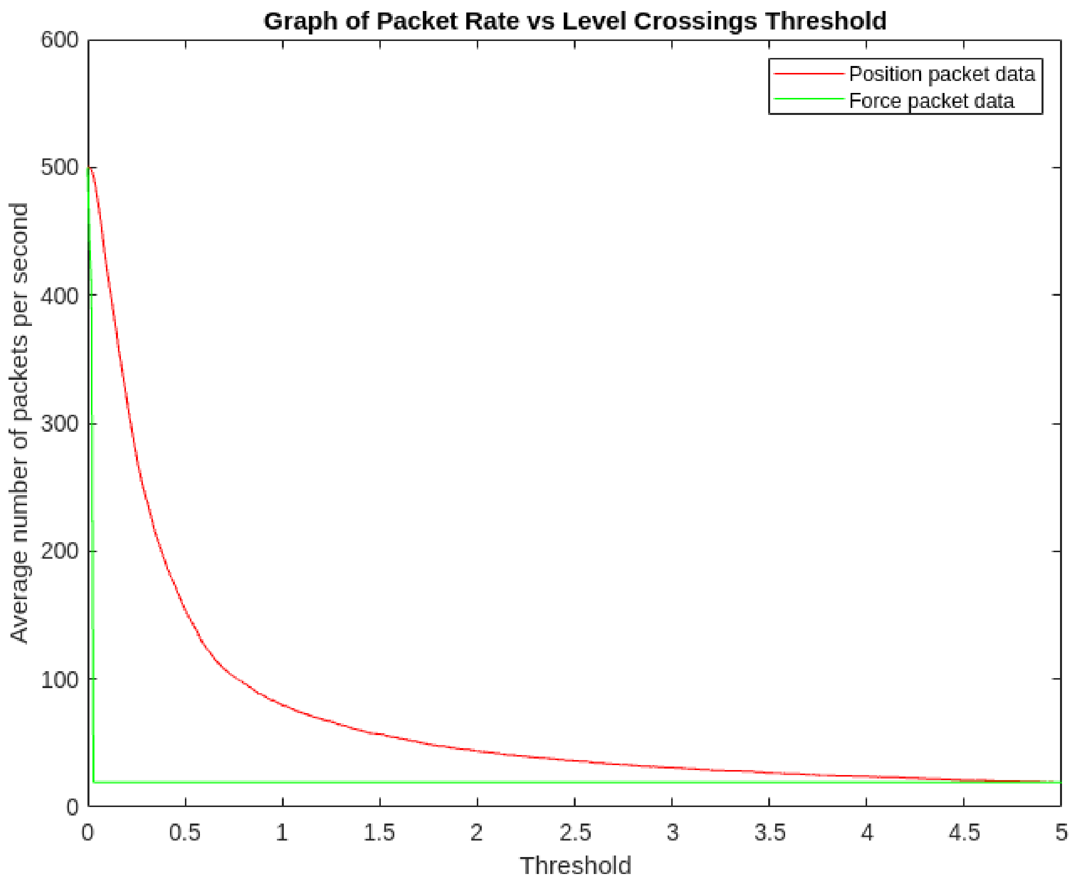
Figure 4.1.1: Weber's Constant vs Packet Rate



in quality, for time periods when no packet is received. The same argument cannot be applied to the graph of Level Crossings threshold as its much lesser reliable. This is because Weber's sampling more or less normalizes the values and hence the change in values.

To understand how the sampling affects the signal, we compared the error between the generated packets and original packets, and it can be seen that a Weber's Sampler gives lower error values. Each packet has multiple attributes and in order to normalize the error values, we take the relative error for each attribute and find the average relative error across attributes. Figure 4.1.3, Figure 4.1.4, Figure 4.1.5 and Figure 4.1.6 show the results in a graphical form. An ideal threshold value for either of the sampling methods is one which results in the minimum error across all threshold

Figure 4.1.2: Level Crossings Threshold vs Packet Rate

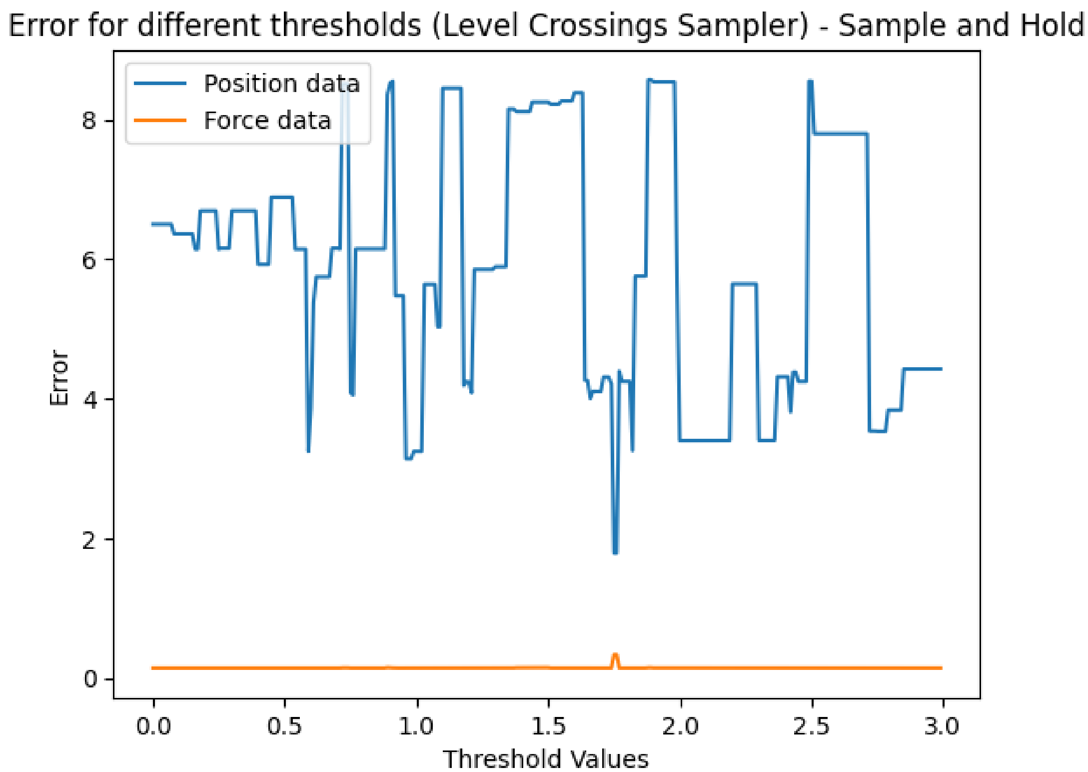


values and can be obtained from these four graphs. Each of these graphs have two plots, one each for position related data and the other for force related data. Since we're considering different threshold values for both, we can pick the threshold value that gives the minimum error rate from each individual plot as the threshold for adaptive sampling.

Clearly, Sample and Hold Extrapolation methods give a very high relative error rate, while also not being very stable. In comparison, the Linear Extrapolation methods give much better results as error rates are significantly reduced. Weber's Sampler gives the least error and should be considered as our primary sampler.

As a Weber's Sampler is more reliable, we would be using it for all our experiments, and consider different threshold values for position and force related data. This is

Figure 4.1.3: Level Crossings Sampling - Sample and Hold



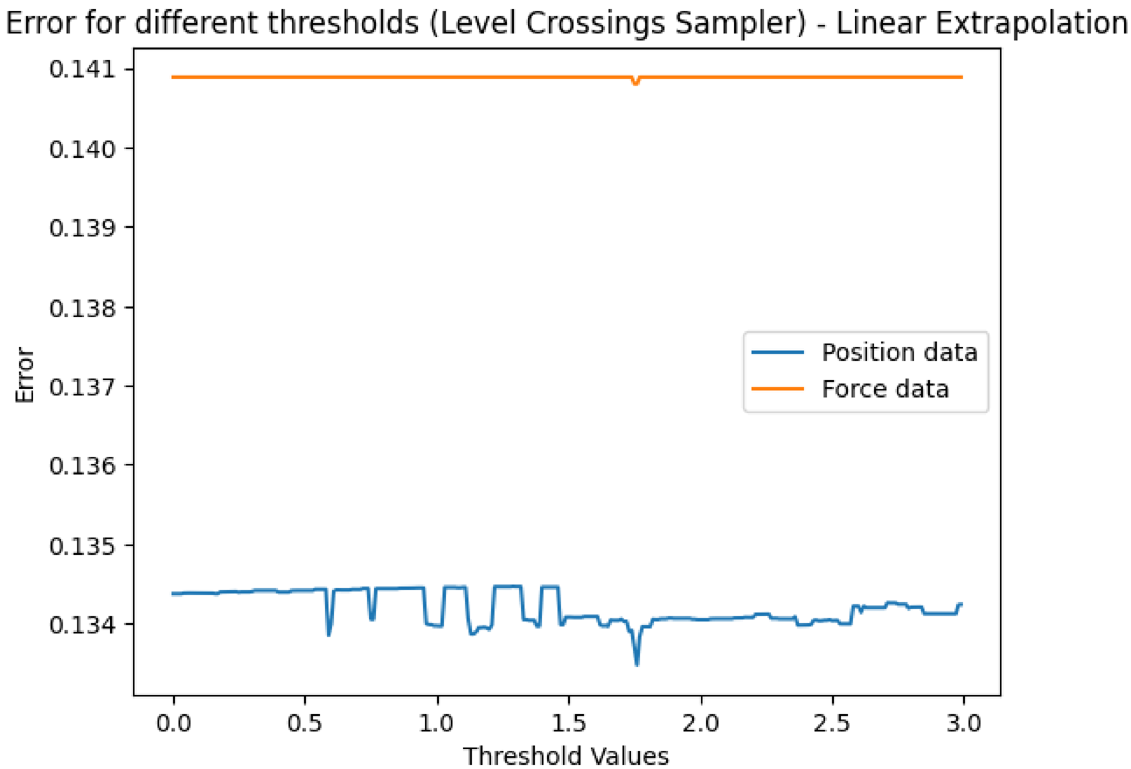
also supported by the results on error variation, because the error hits minimum for Weber's Adaptive Sampling and is a lot more stable with a low error rate.

4.2 Network Design

4.2.1 Work Done

The network simulation has been experimented and implemented in using the OMNet++ simulator. It is is a component-based, modular C++ simulation library and framework designed primarily for the development of network simulators. The INET Framework is a free and open-source model library that works with the OMNeT++ simulation environment. It provides protocols, agents, and other models for communication network researchers and students. INET is especially useful for designing

Figure 4.1.4: Level Crossings Sampling - Linear Extrapolation

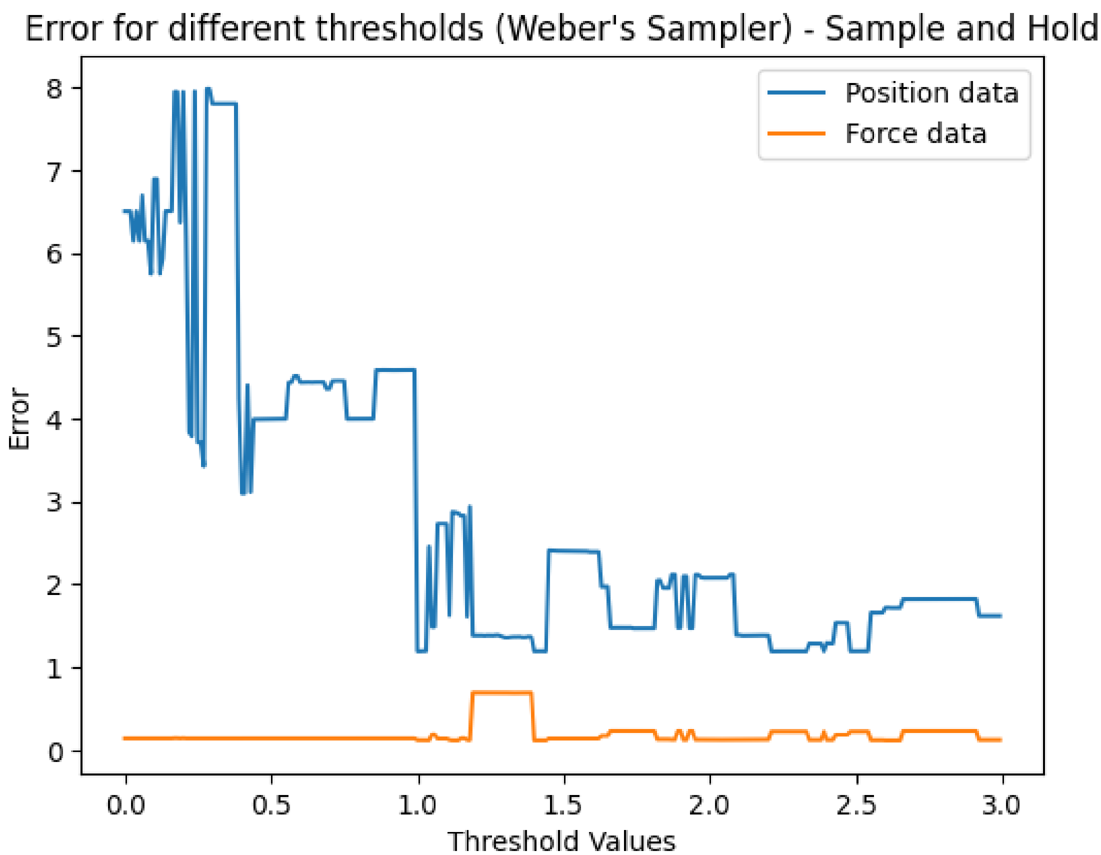


and validating new protocols, as well as investigating novel scenarios [18].

The IEEE 802.1 TSN specific standards are implemented by TSN specific extensions over the OMNet++/INET framefwork for network simulations, and it requires OMNeT++ version 6.0.1 and INET version 4.4.0 under Linux and primarily utilizes upon three types of files - a NED file, a CPP file and an INI file:

- **NED File** - It is a network definition file created by OMNet++.
- **CPP File** - The functionality of the modules defined are implemented in CPP files.
- **INI File** - The configuration file contains options for controlling how the simulation is run.

Figure 4.1.5: Weber's Sampling - Sample and Hold

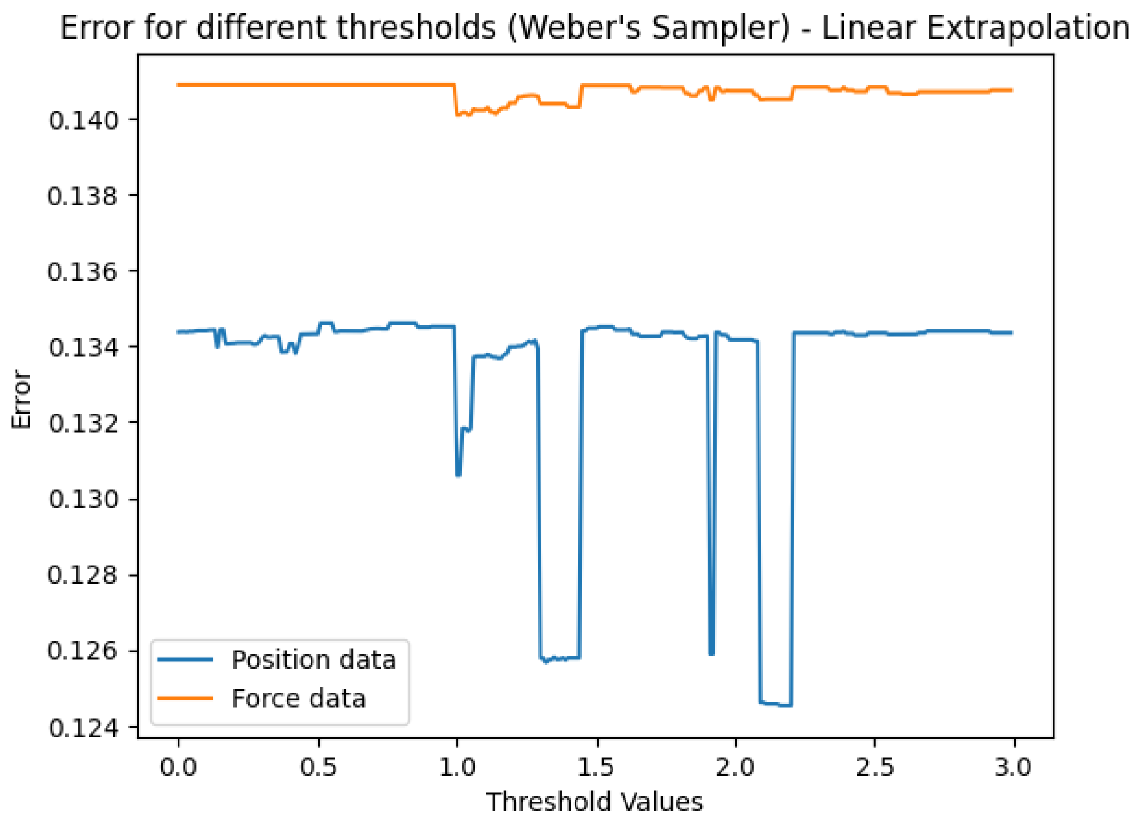


4.2.2 Transmission over the Internet using HoIP

The current examples all are simulated over the same test scenario network which has only one robotic controller (sender) and one robotic arm (receiver). This is a very limited use-case scenario and does not have very realistic applications.

Multiple experiments were done to analyze and improve the performance to reach our expectations. While initially the performance of the networks was not at all upto the mark of the Tactile Internet requirements (end-to-end delay was around 1 second), it was found that after enabling Cut-Through Switching on all the devices, the performance greatly improved and gave acceptable and appreciable results. Cut-through switching is a method for packet switching systems in which the switch begins forwarding a frame (or packet) before the entire frame is received, typically as soon

Figure 4.1.6: Weber's Sampling - Linear Extrapolation



as the destination address and outgoing interface are determined.

Transmission using Three Separate Streams tested over Various Topologies

Below are the various topologies in different testing scenarios as performed. All the simulations were run for 5 seconds, and the default data rate (if not mentioned otherwise) was set as 100Mbps for all links. The default packet size taken is 1000 Bytes. All topologies took 3 applications: Best Effort Haptic Data, Video and Audio. Each of them had their own stream and were assigned Priority Code Point value as 0, 1 and 2. This was assigned in the INI files against every stream associated to the three applications. The results for the various topologies have been analyzed below:

1. **Dumbell Network**

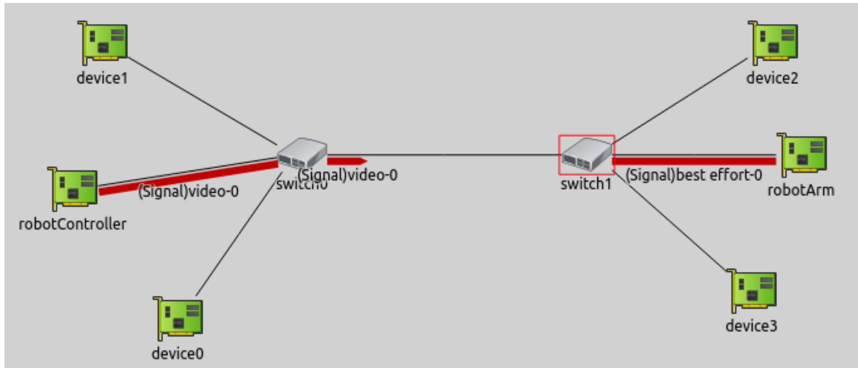


Figure 4.2.1: TSN Dumbell Topology: Dumbell Network with 1 Robot Controller and 1 Robot Arm

The topology built is shown in figure 4.2.1. The simulation was run after enabling cut-through switching, and the end-to-end delay results have been put in table 4.2.1. Further, all the 3 stream delay values accumulated in 5 seconds have been plotted in a graph as shown in figure 4.2.2. From the values documented, it is clear that this topology using TSN is suitable for the tactile internet as the end to end delay is very much under 10 ms.

Robot Controller Application	No. of Packets	Mean Time Taken (in ms)
Best Effort Haptic Data	25191	0.787
Video	16557	0.794
Audio	12683	0.786

Table 4.2.1: TSN Dumbell Topology: Table of Mean of End-to-End Delay Values of the 3 Applications

2. Multiple Routes

The topology built is shown in figure 4.2.3. The simulation was run both without enabling cut-through switching and after enabling, and the end-to-end delay results for have been put in table 4.2.5 and 4.2.3. Further, all the 3 stream delay values accumulated in 5 seconds have been plotted in a graph as shown in figure 4.2.4 (closer look for this in 4.2.5) for without Cut-Through Switching and 4.2.6 for with Cut-Through Switching. From the values documented, it can be observed that without cut-through switching enabled, the number of packets being relayed is significantly lesser on all 3

Figure 4.2.2: TSN Dumbell Topology: End-to-end delay of packet (Mean Bit Life Time per packet) in seconds vs Simulation Time in seconds

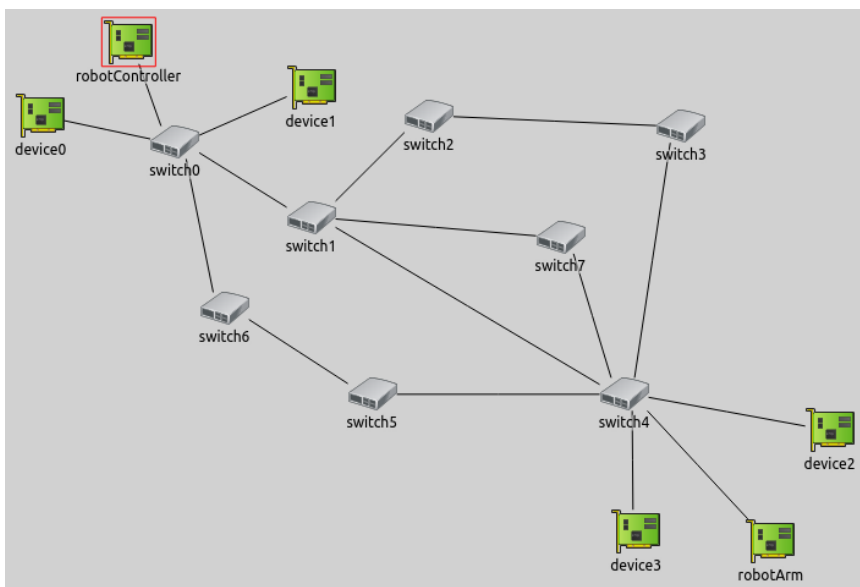


Figure 4.2.3: Multiple Routes Topology: Network with 1 Robot Controller and 1 Robot Arm

streams and the end-to-end delay time is extremely high and does not suit the Tactile Internet requirements. However, with cut-through switching enabled, not only is the number of packets relayed higher, but this topology using TSN is suitable for the tactile internet as the end to end delay is very much under 10 ms.

Robot Controller Application	No. of Packets	Mean Time Taken (in ms)
Best Effort Haptic Data	11540	1351.93
Video	16557	1338.02
Audio	12683	1341.02

Table 4.2.2: Multiple Routes Topology: Table of Mean of End-to-End Delay Values of the 3 Applications without Cut-Through Switching

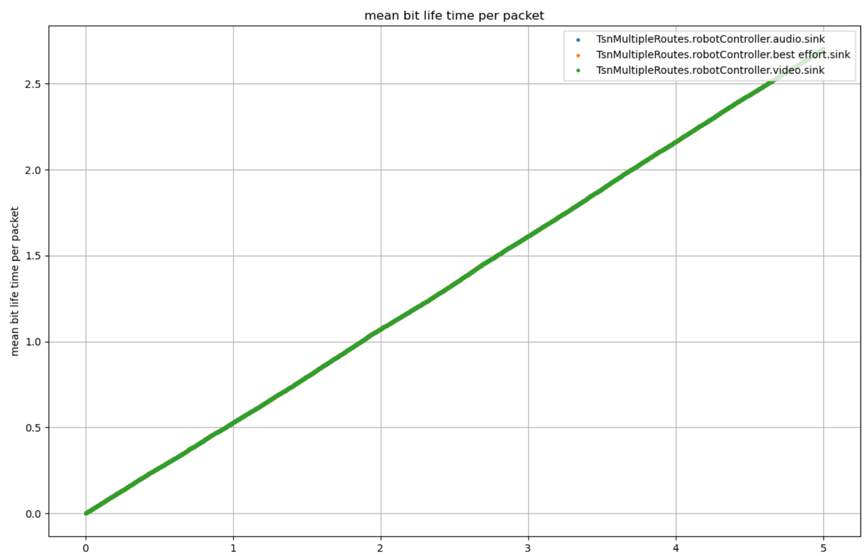


Figure 4.2.4: Multiple Routes Topology: End-to-end delay of packet (Mean Bit Life Time per packet) in seconds vs Simulation Time in seconds without Cut-Through Switching

Robot Controller Application	No. of Packets	Mean Time Taken (in ms)
Best Effort Haptic Data	25191	0.791
Video	16557	0.790
Audio	12683	0.787

Table 4.2.3: Multiple Routes Topology: Table of Mean of End-to-End Delay Values of the 3 Applications with Cut-Through Switch

3. No. of Hops vs Datarate of Links

The topology built is shown in figure 4.2.7. The topology has all enabled cut-through switching on all device and consists of 2 paths from the robot controller to

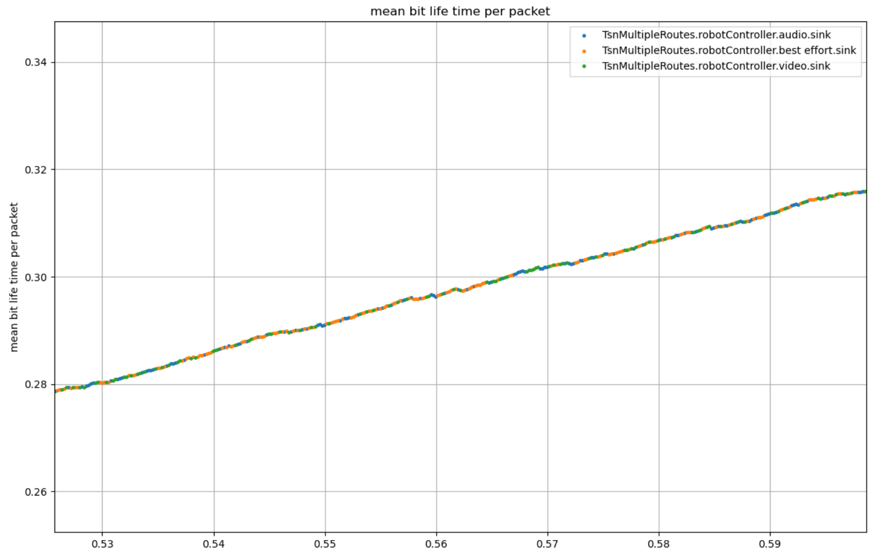


Figure 4.2.5: Multiple Routes Topology: Closeup of End-to-end delay of packet (Mean Bit Life Time per packet) in seconds vs Simulation Time in seconds without Cut-Through Switching

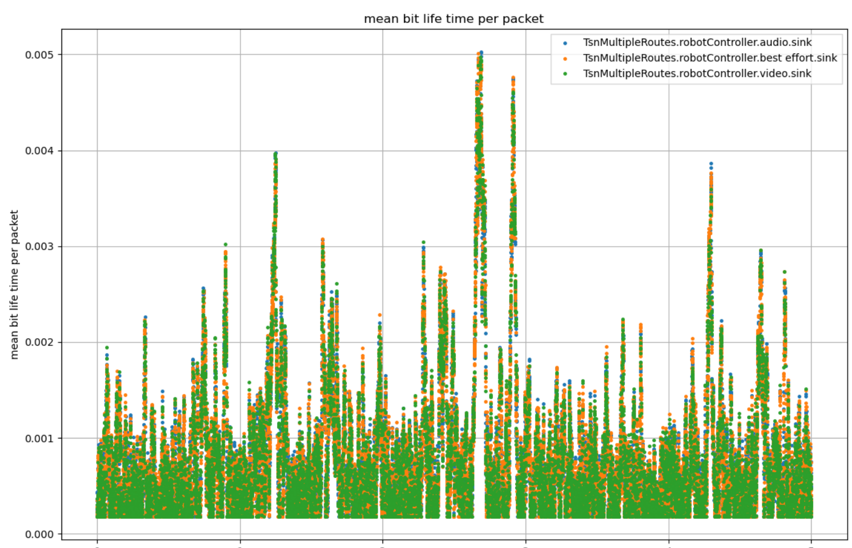


Figure 4.2.6: Multiple Routes Topology: End-to-end delay of packet (Mean Bit Life Time per packet) in seconds vs Simulation Time in seconds with Cut-Through Switching

the robot arm: one having lesser number of hops and lesser datarate of links and the other having more number of hops and higher datarate of links. The packets traversed the path having lesser number of hops and lesser datarate of links.The simulation's

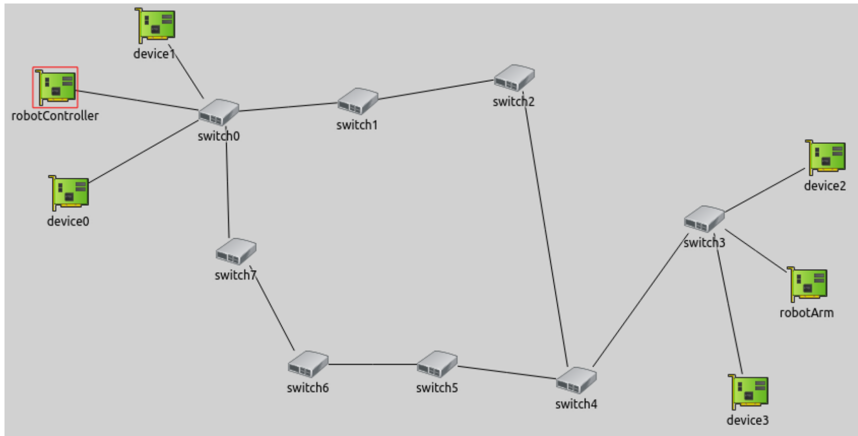


Figure 4.2.7: No. of Hops vs Datarate of Links: 2 Paths differing in No. of Hops and Datarate of Links with 1 Robot Controller and 1 Robot Arm

the end-to-end delay results have been put in table 4.2.4. Further, all the 3 stream delay values accumulated in 5 seconds have been plotted in a graph as shown in figure 4.2.8. From this, it was clarified that the packets will always be transmitted through the path having lesser number of hops regardless of others conditions.

Robot Controller Application	No. of Packets	Mean Time Taken (in ms)
Best Effort Haptic Data	25191	0.798
Video	12683	0.797
Audio	16557	0.805

Table 4.2.4: No. of Hops vs Datarate of Links: Table of Mean of End-to-End Delay Values of the 3 Applications

4. Cut-Through Path or Lesser Number of Hops Path

The topology built is shown in figure 4.2.9. The topology consists of 2 paths from the robot controller to the robot arm: one having lesser number of hops but no cut-through switching enabled and the other having cut-through switching enabled. The packets nonetheless still traversed the path having lesser number of hops and no cut-through switching, as expected, and the simulation's the end-to-end delay results have been put in table 4.2.1. Further, all the 3 stream delay values accumulated in 5 seconds have been plotted in a graph as shown in figure 4.2.10 (it is similar values as multiple routes topology without cut through switching). From the values documented, it is clear that this topology without cut-through switching enabled

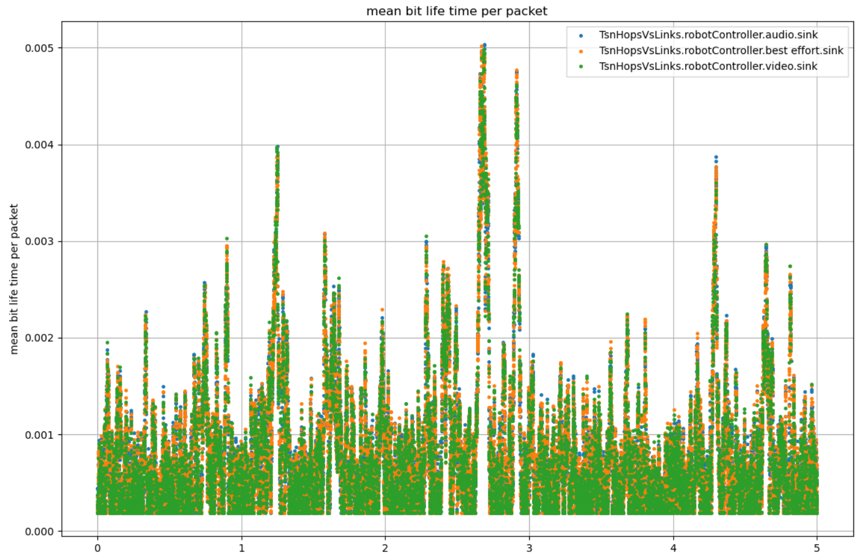


Figure 4.2.8: No. of Hops vs Datarate of Links: End-to-end delay of packet (Mean Bit Life Time per packet) in seconds vs Simulation Time in seconds

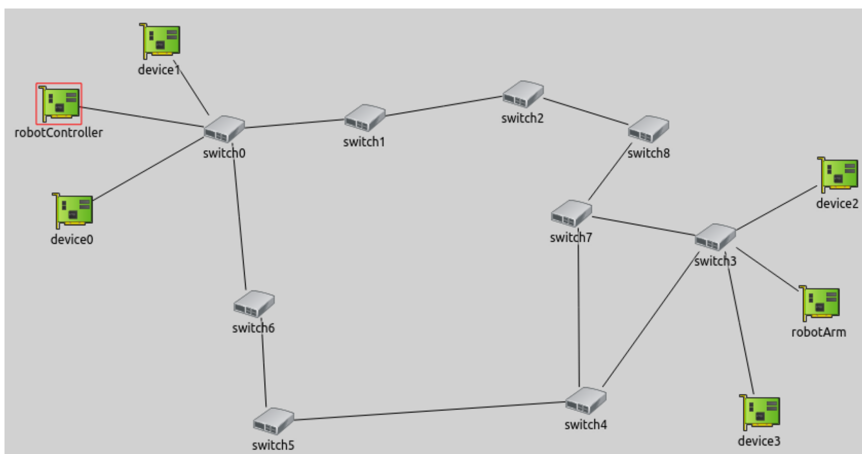


Figure 4.2.9: Cut-Through vs No. of Hops: 2 Paths differing in No. of Hops and Cut-Through Path with 1 Robot Controller and 1 Robot Arm

everywhere is not suitable for the tactile internet as the end to end delay is very high.

5. Multiple Robot Controllers and Robot Arms

The topology built is shown in figure 4.2.11. The simulation was run with cut-through switching, and the end-to-end delay results have been put in table 4.2.6 and 4.2.7 for both the robot pairs respectively. Further, all the 3 stream delay values accumulated in 5 seconds have been plotted in a graph as shown in figure 4.2.12 and 4.2.13 respectively. From the values documented, it is clear that multiple haptic

Robot Controller Application	No. of Packets	Mean Time Taken (in ms)
Best Effort Haptic Data	11539	1352.16
Video	5867	1338.12
Audio	7610	1341.13

Table 4.2.5: Cut-Through vs No. of Hops: Table of Mean of End-to-End Delay Values of the 3 Applications

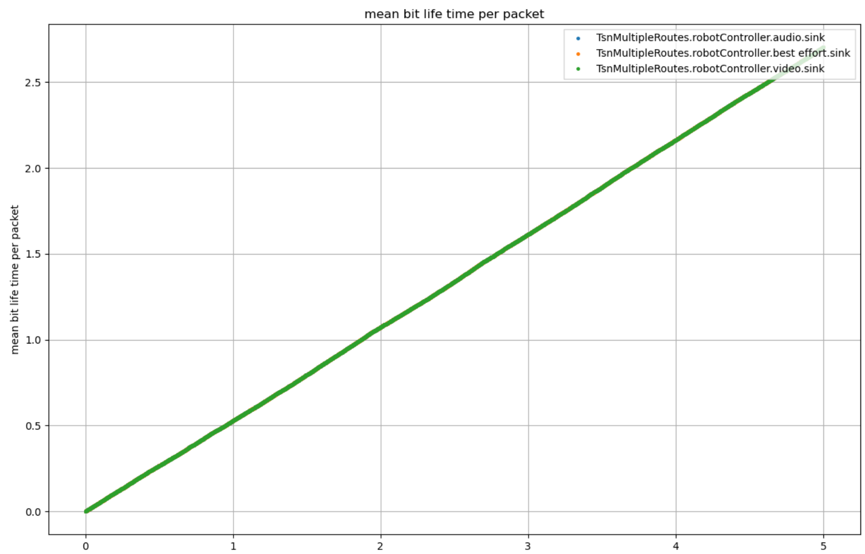


Figure 4.2.10: Cut-Through vs No. of Hops: End-to-end delay of packet (Mean Bit Life Time per packet) in seconds vs Simulation Time in seconds

devices are supported provided they have unique receivers, and thus this topology using TSN is suitable for the tactile internet as the end to end delay is very much under 10 ms. If there are multiple haptic devices sending data to a single receiver, then cut-through switching is not possible and from previous experiments, we can see that communication done without cut-through switching does not satisfy the tactile internet end-to-end delay requirement.

From the results of the 5 topologies simulations, it can be understood that Cut-Through Switching enabled helps in ensuring the haptic communication supports the Tactile Internet constraint of less than 10ms delay.

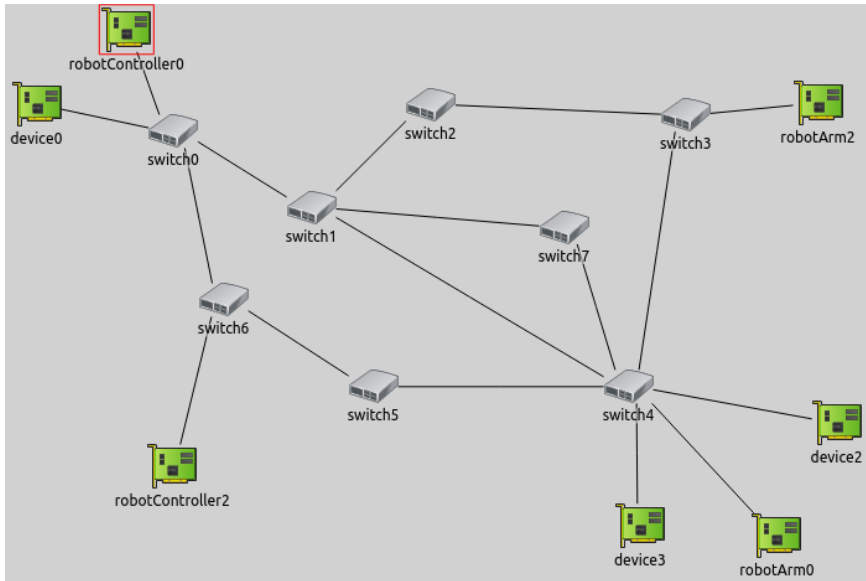


Figure 4.2.11: Multiple Robot Pairs: Multiple Paths with 2 Robot Controller and 2 Robot Arm

Robot Controller Application	No. of Packets	Mean Time Taken (in ms)
Best Effort Haptic Data	25112	0.731
Video	12539	0.727
Audio	16625	0.733

Table 4.2.6: Multiple Robot Pairs: Robot Controller 0 - Table of Mean of End-to-End Delay Values of the 3 Applications

Robot Controller Application	No. of Packets	Mean Time Taken (in ms)
Best Effort Haptic Data	24964	0.761
Video	12750	0.763
Audio	16860	0.765

Table 4.2.7: Multiple Robot Pairs: Robot Controller 2 - Table of Mean of End-to-End Delay Values of the 3 Applications

Transmission by augmenting Haptic Data along with Media Data tested over Multiple Routes Topology

The simulations were all run on a single topology as is shown in figure 4.2.3. All the simulations were run for 5 seconds. There were 3 applications: Best Effort Hap-

Figure 4.2.12: Multiple Robot Pairs: RobotController 0 - End-to-end delay of packet (Mean Bit Life Time per packet) in seconds vs Simulation Time in seconds

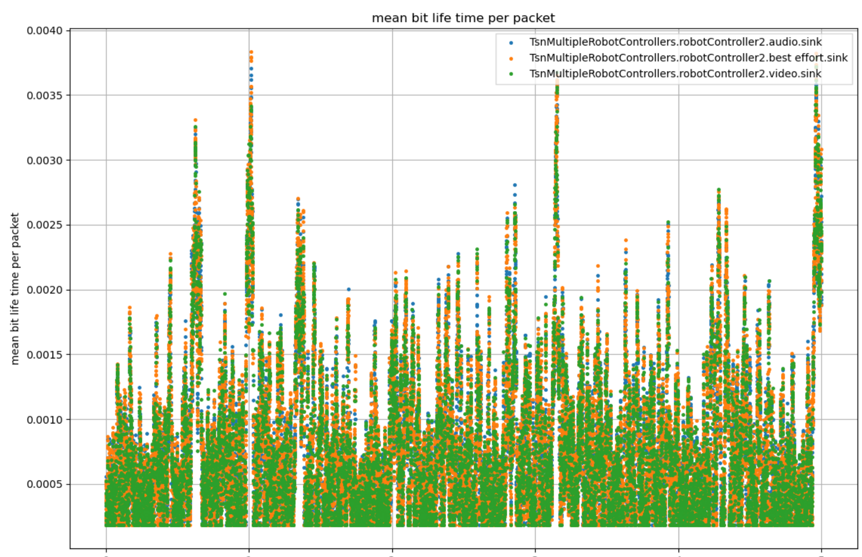


Figure 4.2.13: Multiple Robot Pairs: RobotController 2 - End-to-end delay of packet (Mean Bit Life Time per packet) in seconds vs Simulation Time in seconds

tic Data, Video and Audio. Each of them had their own stream and were assigned Priority Code Point value as 0 all. This was assigned in the INI files against every stream associated to the three applications. The simulation was run both without enabling cut-through switching and after enabling. The 8 simulations run varying the channel datarates and the size of data packets being sent across. The datarates and size of data packets being sent were both varied between 116 Bytes (size of only haptic data packet) and 1500 Bytes (maximum ethernet packet transmission size - size of haptic+audio and haptic+video packet).

1. Varying Data Packet Size With Constant Data Packet Size of 116 Bytes

The simulation was run setting constant data packet size to 116 Bytes (Only Haptic Packet Data), which is the size of only Haptic Packet Data, and changing the channel data rates between 1500 Bytes (Maximum ethernet frame transmission size - size of haptic+audio and haptic+video packet) and 116 Bytes. Further, the simulations were run by enabling both Cut-Through Switching and without Cut-Through Switching (Storing and Forwarding Packets). The results with Cut-Through Switching can be found in Table 4.2.8 and without Cut-Through Switching can be found in Table 4.2.9.

Channel Data Rate	No. of Packets	Mean Time Taken (in ms)
116 Bytes	25193	0.003
1500 Bytes	21418	0.034

Table 4.2.8: Varying Channel Data Rate with 116 Bytes Data Packet Size with Cut Through Switching Enabled

Channel Data Rate	No. of Packets	Mean Time Taken (in ms)
116 Bytes	7062	0.008
1500 Bytes	25193	0.071

Table 4.2.9: Varying Channel Data Rate with 116 Bytes Data Packet Size without Cut Through Switching Enabled

It can be inferred from the results that when it's a data packet of only haptic data (size of 116 bytes), the Tactile Internet delay constraint of less than 10ms is met regardless of the channel rates being varied between 116 bytes and 1500 bytes. Further, while the simulation with enabled Cut-Through Switching has better results than Store and Forward (as expected), both fulfill the criteria to support Tactile Internet. The main reason for this being the channel data rate being equal to or higher than the set data packet size.

2. Varying Channel Data Rates With Constant Data Packet Size of 1500 Bytes

The simulation was run setting constant data packet size to 1500 Bytes (Ethernet Transmission limit), which is the size of both Haptic+Audio packets and Haptic+Video packets, and changing the channel data rates between 1500 Bytes and 116 Bytes (116 Bytes is the size of only haptic packet data size). Further, the simulations were run by enabling both Cut-Through Switching and without Cut-Through Switching (Storing and Forwarding Packets). The results with Cut-Through Switching can be found in Table 4.2.10 and without Cut-Through Switching can be found in Table 4.2.11.

Channel Data Rate	No. of Packets	Mean Time Taken (in ms)
116 Bytes	21418	381.48
1500 Bytes	25193	0.018

Table 4.2.10: Varying Channel Data Rate with 1500 Bytes Data Packet Size with Cut Through Switching Enabled

Channel Data Rate	No. of Packets	Mean Time Taken (in ms)
116 Bytes	7062	1787.08
1500 Bytes	24964	0.108

Table 4.2.11: Varying Channel Data Rate with 1500 Bytes Data Packet Size without Cut Through Switching Enabled

It can be inferred from the results that when it's a data packet of only haptic data (size of 1500 bytes), the Tactile Internet delay constraint of less than 10ms is met only when the datarate is set to 1500 bytes with Cut Through Switching and Store & Forward Switching. For a channel datarate of 116 bytes, the 1500 byte packet size transmission takes much longer with both Cut Through Switching and Store and

Forward switching, going upto even 2 seconds. This is clearly not under the 10ms requirement for the Tactile Internet, and thus if a datarate is set to only Haptic Data Packet size, then the augmentation of Haptic with Audio/Video packets will actually be counter-effective for the tactile internet. The main reason for this being the channel data rate being almost 10 times lesser than the set data packet size.

From the results of the 8 simulations, we can understand that as long as the data rate is equal to or greater than the size of the data packets (116 Bytes data rate if only haptic, 1500 bytes if haptic, audio and video data is involved), the augmentation method for transmission of haptic data is efficient in utilizing the channel to the fullest.

These results show that both the methods of transmission of the haptic packets over the internet (through 3 separate datastreams and through augmentation of the media data along with haptic data) are good ways for maintaining the Tactile Internet requirement of less than 10ms delay while transmitting haptic packets over the internet.

4.3 Routing Algorithm

The modified shortest path algorithm incorporating the history of delays across routes has been implemented and a run-through of the algorithm in a topology is given below. Figure 4.3.1 represents the topology. This has four paths from the robot controller to the robotic arm, as shown in Figure 4.3.2, Figure 4.3.3, Figure 4.3.4 and Figure 4.3.5.

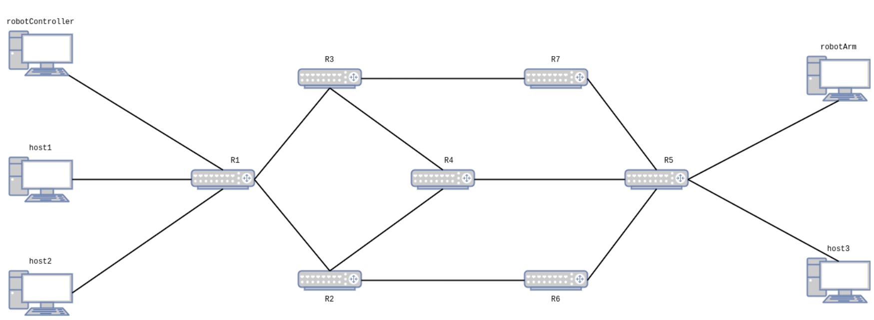


Figure 4.3.1: Sample Topology with four paths from source to destination

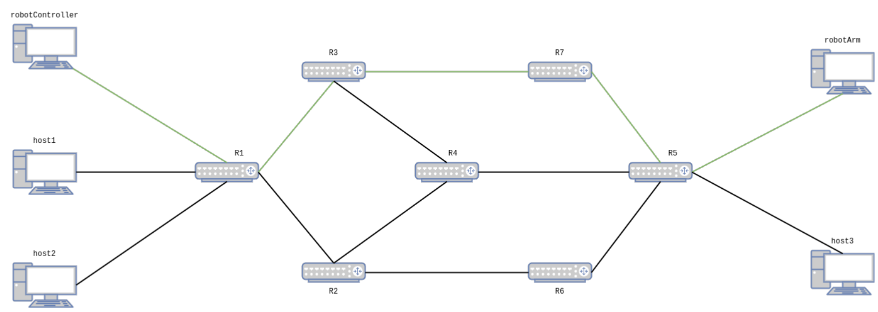


Figure 4.3.2: First path from robot controller to robotic arm

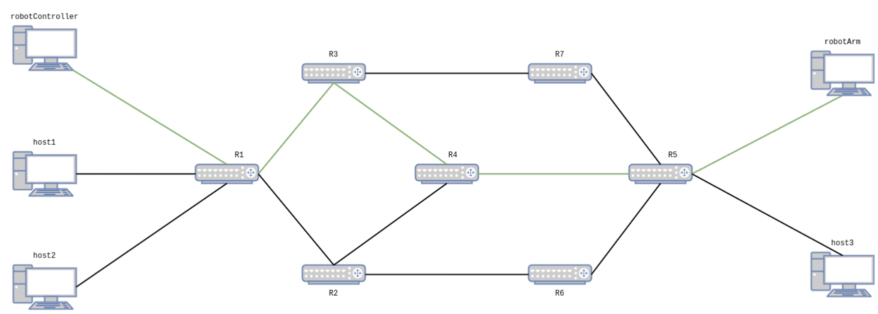


Figure 4.3.3: Second path from robot controller to robotic arm

The algorithm uses two tables, that stores average delays between source and destination recorded at specified time intervals throughout the day. We are considering intervals of one hour each to record path delays. While one table stores the delay in each edge, the other is derived from the first table and stores the end-to-end delay between source and destination across different paths. Considering the topology given in Figure 4.3.1, the table below depicts the hourly delays in each edge.

4.3.1 Working of the Modified Shortest Path Algorithm

Table 4.3.1 shows only some of the values stored. In fact, all source destination pairs have 24 records in the table, one for each hour in the day. Since the topology has 14 edges, the history table would have 14 * 24 = 336 records. The history table is not the same as the regular IPv4 routing table for a router, even though they both store similar values, since this table is only used for haptic communications and there are additional parameters like time and delay that aren't required in a routing table.

This table is used in the shortest path algorithm to calculate edge weights. Edge weights calculate the average delay in the edge throughout the day and consider the

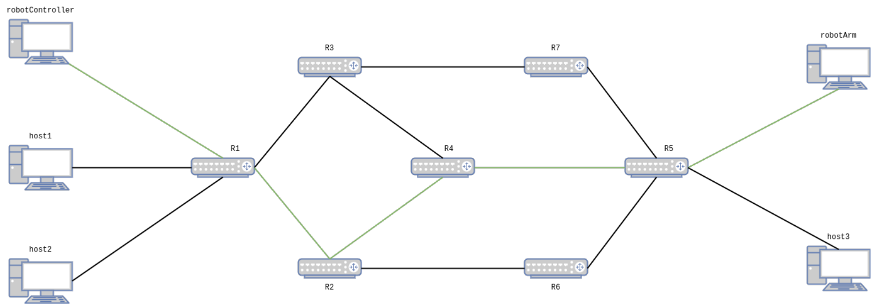


Figure 4.3.4: Third path from robot controller to robotic arm

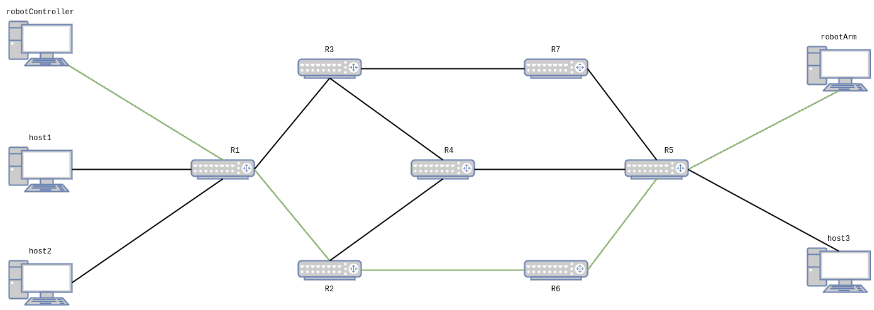


Figure 4.3.5: Fourth path from robot controller to robotic arm

Source	Destination	Time (Hours)	Delay (in ms)
robotController	R1	00	1ms
robotController	R1	01	1ms
robotController	R1	02	1ms
robotController	R1	03	3ms
robotController	R1	04	3ms
robotController	R1	05	3ms
..	..	..	..
R1	R2	00	1ms
R1	R2	01	1ms
R1	R2	02	2ms
..	..	..	..

Table 4.3.1: Sample values from the history table storing the delay in each edge, recorded every hour, in the network

one with the least delay and least variation in the delay. From the table, the average delay between *robotController* and *R1* is,

$$\mu = \frac{\Sigma x_i}{n} = \frac{1+1+1+3+3+3+....}{24} = 1.67ms$$

Similarly, the standard deviation can be calculated for all the recorded values as,

$$\sigma = \sqrt{\frac{\Sigma(x_i-\mu)^2}{n-1}} = \sqrt{\frac{(1-1.67)^2+(1-1.67)^2+...}{23}} = 0.80ms$$

Let β be equal to 1, since this gives a good measure of average delay along with its steadiness, from experiments. Therefore, edge weight for the same can be calculated as,

$$edge\_weight = \mu + \beta * \sigma = 1.67 + 0.80 = 2.47ms$$

Similarly, the edge weight can be calculated for all edges and shortest path out of all can be considered for communication.

4.3.2 Working of the Modified RSVP Algorithm

The modified RSVP algorithm checks if the shortest path obtained previously is suitable for haptic communication. If not, it checks for the next shortest path and so on until it finds a suitable path. Table 4.3.2 contains average delays between source and destination recorded every hour throughout the day, across different paths.

Source	Destination	Path	Time (Hours)	Delay (in ms)
robotController	robotArm	Path 1	00	12.831ms
robotController	robotArm	Path 2	00	8.856ms
robotController	robotArm	Path 3	00	5.662ms
robotController	robotArm	Path 4	00	5.795ms
robotController	robotArm	Path 1	01	8.856ms
robotController	robotArm	Path 2	01	9.057ms
..	..	..	..	..
robotController	robotArm	Path 2	18	9.056ms
robotController	robotArm	Path 3	18	6.662ms
robotController	robotArm	Path 4	18	9.060ms
..	..	..	..	..

Table 4.3.2: Sample values from the history table storing the delay in each path, recorded every hour, in the network

The table stores the recorded end-to-end delay values between robotic controller and robotic arm in each path. The values can be seen to be slightly greater than sum of delays in each edge and this is because of processing delays at each node. Table 4.3.2 shows only some of the values stored. In fact, all paths with source as a robotic controller and destination as a robotic arm have 24 records in the table, one for each hour in the day. Since the topology has four paths from the robotic controller to the arm, the path history table would totally have 4 * 24 = 96 records.

This table is used in the RSVP algorithm to check whether a dedicated path between the source and destination can be reserved. From the table, the average delay in Path 2 during the day is,

$$\mu = \frac{\Sigma x_i}{n} = \frac{8.856 + 9.057 + 9.056 + ...}{24} = 8.66ms$$

Similarly, the standard deviation can be calculated for all the recorded values as,

$$\sigma = \sqrt{\frac{\Sigma(x_i - \mu)^2}{n-1}} = \sqrt{\frac{(8.856 - 8.66)^2 + (9.057 - 8.66)^2 + ...}{23}} = 0.76ms$$

Clearly, $\mu + \sigma = 8.66 + 0.76 = 9.42ms$ is lesser than our threshold, γ, which is 10ms. Considering the constant, α, to be 2ms, we can see that the variance does not cross it and the delay in the path is steady. Therefore, according to Algorithm 4.3.2, haptic communication will not be disabled, and the path would be reserved if it's the shortest path.

Chapter 5

CONCLUSION AND FUTURE WORK

In this work, we have analysed previous work and developed a method for haptic communication over the Tactile Internet. In doing so, we have standardized the mechanism to represent haptic information for communication over the network. A method to optimize the generation of haptic packets and reduce error at the receiver has also been explored. Further, different methods to prioritize haptic communication in the network were studied and implemented. Finally, a routing algorithm to support haptic communication over the Internet has been designed and developed.

In addition to this, this literature will help increase the existing literature and implementation on Time Sensitive Networking in the Tactile Internet and increase proposed enhancements of existing standards for communication to support the tactile internet.

Currently, actual haptic data has been obtained as a dataset and has had adaptive sampling done on it. A comparison of different types of sampling has also been done, and an efficient structure has been developed for haptic data representation. For prioritizing haptic data, two methods have been identified: one based on priority code point assignment and one based on augmentation of audio/video data onto the haptic data. Both have been simulated over various scenarios, and the experiments have shown that the former method implemented with cut-through switching, it is suitable for supporting the tactile internet (including having multiple robotic pairs). The latter supports the tactile internet provided the packet size being sent across is less than or equal to the channel datarates set.

A routing algorithm to support haptic communication over the Tactile Internet has been proposed. This is a modified version of the shortest path algorithm based on delay, and a history of delays on each path has been considered which makes this work unique, and ensures that the selected route is always steady and does not have a constantly changing packet delay. Further, a modified version of the RSVP algorithm has been used to ensure that the bandwidth is allocated as required for communication.

Future work includes coming up with a detailed way of processing the haptic packet on the receiver end after being passed through the network. Further, augmentation method is to be applied to prioritize haptic data, and this is to be experimented with and compared to the results of plain priority code point assignment. The routing protocol can also be improved by expanding the history table and using an efficient way to store and retrieve data from the table.

REFERENCES

[1] V. Gavrilut, A. Pruski, M. S. Berger, "Constructive or Optimized: An Overview of Strategies to Design Networks for Time-Critical Applications", Proceedings of the ACM Computing Surveys, vol. 55, no. 3, article 62, 2022, doi: 10.1145/3501294

[2] D. Van Den Berg et al., "Challenges in Haptic Communications Over the Tactile Internet," in IEEE Access, vol. 5, pp. 23502-23518, 2017, doi: 10.1109/ACCESS.2017.2764181.

[3] Z. Cao, Q. Liu, D. Liu and Y. Hu, "Enhanced System Design and Scheduling Strategy for Switches in Time-Sensitive Networking," in IEEE Access, vol. 9, pp. 42621-42634, 2021, doi: 10.1109/ACCESS.2021.3061969.

[4] N. Finn, "Introduction to Time-Sensitive Networking," in IEEE Communications Standards Magazine, vol. 2, no. 2, pp. 22-28, JUNE 2018, doi: 10.1109/MCOMSTD.2018.1700076.

[5] A. B. D. Kinabo, J. B. Mwangama and A. A. Lysko, "Towards Wi-Fi-based Time Sensitive Networking Using OMNeT++/NeSTiNg Simulation Models," 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), 2021, pp. 1-6, doi: 10.1109/ICECET52533.2021.9698580.

[6] Ahmed Nasrallah, Venkatraman Balasubramanian, Akhilesh S. Thyagaturu, Martin Reisslein, Hesham ElBakoury, "Reconfiguration Algorithms For High Precision Communication in Time-Sensitive Networks: Time-Aware Shaper Configuration With IEEE 802.1QCC" ITU Journal on Future and Evolving Technologies, Volume 2 (2021), Issue 1, 15 March 2021

[7] A. Alnajim, S. Salehi and C. -C. Shen, "Incremental Path-Selection and Scheduling for Time-Sensitive Networks," 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9013427.

[8] Nattakorn Promwongsa, Amin Ebrahimzadeh, Diala Naboulsi, Somayeh Kianpisheh, Fatna Belqasmi, "A Comprehensive Survey of the Tactile Internet:

State-of-the-art and Research Directions". Communications Surveys and Tutorials, IEEE Communications Society, Institute of Electrical and Electronics Engineers, 2020, ff10.1109/COMST.2020.3025995ff. ffhal-02963827

[9] Delia Rico, Maria-del-Mar Gallardo, and Pedro Merino. 2021. Modeling and verification of the Multi-connection Tactile Internet Protocol. In Proceedings of the 17th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '21). Association for Computing Machinery, New York, NY, USA, 105–114. https://doi.org/10.1145/3479242.3487328

[10] Marshall, Alan Yap, Kian Meng Yu, Wai. (2008). Providing QoS for Networked Peers in Distributed Haptic Virtual Environments. Advances in Multimedia. 2008. 10.1155/2008/841590.

[11] Podlesny, Maxim, "Networking Mechanisms for Delay-Sensitive Applications" (2009). All Theses and Dissertations (ETDs). 278. https://openscholarship.wustl.edu/etd/278

[12] Subarna Singh (2017), "Routing Algorithms for Time Sensitive Networks", Master Thesis, Institute of Parallel and Distributed Systems

[13] K. Huang, J. Wu, X. Jiang, D. Xiong, H. Yao, W. Xu, Y. Peng, Z. Liu, "A Period-Aware Routing Method for IEEE 802.1Qbv TSN Networks", Electronics 2021, 10, 58. https://doi.org/10.3390/electronics10010058

[14] J. Falk et al., "NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++," 2019 International Conference on Networked Systems (NetSys), 2019, pp. 1-8, doi: 10.1109/NetSys.2019.8854500.

[15] V. Gokhale, O. Dabeer and S. Chaudhuri, "HoIP: Haptics over Internet Protocol," 2013 IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE), Istanbul, Turkey, 2013, pp. 45-50, doi: 10.1109/HAVE.2013.6679609.

[16] V. Gokhale, S. Chaudhuri and O. Dabeer, "HoIP: A point-to-point haptic data communication protocol and its evaluation," 2015 Twenty First National

Conference on Communications (NCC), Mumbai, India, 2015, pp. 1-6, doi: 10.1109/NCC.2015.7084908.

[17] Wikipedia. Time Sensitive Networking. https://en.m.wikipedia.org/wiki/Time-Sensitive_Networking

[18] INET. What is INET Framework? https://inet.omnetpp.org/Introduction.html

BIODATA

- **Gaurang Jitendra Velingkar**
  191IT113
  gaurangvelingkar@gmail.com
  +919845031419
  G-201, Keerthi Harmony, 4th Main Raghavendra Nagar, Behind Koshys Hospital, T.C. Palya Main Road, Ramamurthy Nagar, Bangalore, Karnataka, India - 560016

- **Rakshita Varadarajan**
  191IT140
  rakshitajps@gmail.com
  +919148141807
  No. 36, 16th Cross, 6th Phase JP Nagar, Bengaluru, Karnataka, India - 560078

- **Stafan Kuttikal Santhosh**
  191IT151
  stafansanthosh@gmail.com
  +918147445130
  C—o Kuttikal Puthenpurayil, Purakkad, Ambalapuzzha, Kerala, India - 688561

IT report

ORIGINALITY REPORT

10%	7%	7%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	1%
2	Submitted to National Institute of Technology Karnataka Surathkal Student Paper	1%
3	cse.wustl.edu Internet Source	1%
4	"Communication in Critical Embedded Systems", Springer Science and Business Media LLC, 2017 Publication	1%
5	d-nb.info Internet Source	1%
6	repository.tudelft.nl Internet Source	<1%
7	Oliver Holland, Eckehard Steinbach, R. Venkatesha Prasad, Qian Liu et al. "The IEEE 1918.1 “Tactile Internet” Standards Working Group and its Standards", Proceedings of the IEEE, 2019	<1%

Publication

8	en.wikipedia.org Internet Source	<1 %
9	www.hindawi.com Internet Source	<1 %
10	backend.orbit.dtu.dk Internet Source	<1 %
11	repository.up.ac.za Internet Source	<1 %
12	www.journaltocs.ac.uk Internet Source	<1 %
13	Arnold B. D. Kinabo, Joyce B. Mwangama, Albert A. Lysko. "Towards Wi-Fi-based Time Sensitive Networking Using OMNeT++/NeSTiNg Simulation Models", 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), 2021 Publication	<1 %
14	B. Ya. Likhttsinder, Yu. O. Bakai. "Ethernet networks with deterministic delays", AIP Publishing, 2023 Publication	<1 %
15	zenodo.org Internet Source	<1 %

16 Madan Kumar S, Raja Vara Prasad Y, Paventhan A. "Latency-Aware Tactile Cyber-Physical Systems for Mission-Critical Applications", 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2022 <1%
Publication

17 "The Tactile Internet", Wiley, 2021 <1%
Publication

18 Nattakorn Promwongsa, Amin Ebrahimzadeh, Diala Naboulsi, Somayeh Kianpisheh et al. "A Comprehensive Survey of the Tactile Internet: State-of-the-Art and Research Directions", IEEE Communications Surveys & Tutorials, 2021 <1%
Publication

19 Simos, Dimitrios, Ioannis Papaefstathiou, and Manolis G.H Katevenis. "Building an FoC Using Large, Buffered Crossbar Cores", IEEE Design & Test of Computers, 2008. <1%
Publication

20 "Networking 2000 Broadband Communications, High Performance Networking, and Performance of Communication Networks", Springer Science and Business Media LLC, 2000 <1%
Publication

21	Liqiang Zhang, Sherali Zeadally. "Enabling End-to-End QoS over Hybrid Wired-Wireless Networks", Wireless Personal Communications, 2006 Publication	<1%
22	www2.math.uu.se Internet Source	<1%
23	Christoph Meinel, Harald Sack. "Internetworking", Springer Science and Business Media LLC, 2013 Publication	<1%
24	Jonathan Falk, David Hellmanns, Ben Carabelli, Naresh Nayak, Frank Durr, Stephan Kehrer, Kurt Rothermel. "NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++", 2019 International Conference on Networked Systems (NetSys), 2019 Publication	<1%
25	Moonbeom Kim, Doyeon Hyeon, Jeongyeup Paek. "eTAS: enhanced Time-Aware Shaper for Supporting Non-Isochronous Emergency Traffic in Time-Sensitive Networks", IEEE Internet of Things Journal, 2021 Publication	<1%
26	pta-dspace-dmz.csir.co.za Internet Source	<1%
27	researchspace.csir.co.za	

Internet Source

<1%

28 www.slideshare.net
Internet Source

<1%

29 www.ee.iitb.ac.in
Internet Source

<1%

30 Kurian Polachan, Chandramani Singh, T. V. Prabhakar. "Decentralized Dynamic Scheduling of TCPS Flows and a Simulator for Time-sensitive Networking", ACM Transactions on Internet Technology, 2022
Publication

<1%

31 Tingrong Lu, Chengcheng Sui, Yushu Ma, Jinsong Zhao, Yongtian Yang. "Chapter 40 Extending Address Space of IP Networks with Hierarchical Addressing", Springer Science and Business Media LLC, 2005
Publication

<1%

32 "Autonomous and Connected Vehicles", Wiley, 2022
Publication

<1%

33 Sarah Pila, Alexis R. Lauricella, Anne Marie Piper, Ellen Wartella. "The power of parent attitudes: Examination of parent attitudes toward traditional and emerging technology", Human Behavior and Emerging Technologies, 2021

<1%

Publication

34	Toni Adame, Marc Carrascosa-Zamacois, Boris Bellalta. "Time-Sensitive Networking in IEEE 802.11be: On the Way to Low-Latency WiFi 7", Sensors, 2021 Publication	<1%
35	www.lia.ufc.br Internet Source	<1%

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		

Department of Information Technology
National Institute of Technology Karnataka, Surathkal-575025

Permission from Major Project-II Guide to Appear for End Semester Evaluation of Major Project-II (IT499)

We students of 8th semester B.Tech.(IT) carried out the Major Project-II titled "*Design and Simulation of Haptic Communication over the Tactile Internet in a Time-Sensitive Environment*" under your guidance of *Dr. Geetha V* from 02nd Jan 2023 to 24th April 2023. We have shown the Major Project-II progress regularly to my major project guide and incorporated all technical suggestions given by my/our Major Project guide in the Major Project-II.

Progress of the our Major Project-II in points-wise are as follows:

1. Mechanism to represent haptic information for communication over the network has been devised.
2. 2 Methods have been developed to prioritize haptic communication in the existing network.
3. A routing algorithm to support Tactile Internet in the existing network has been designed.

Gaurang Jitendra Velingkar (191IT113) 07/04/23

Rakshita Varadarajan (191IT140) 7/4/23

Stafan Kuttikal Santhosh (191IT151) 7/04/2023

Aforesaid students have shown the progress as well as Report of the Major Project-II (IT499) carried out by them. Progress is satisfactory. (If progress is not satisfactory stickout this line and write the comments below)

Write comments/remarks if the progress of Major Project-II (IT499) is not satisfactory

----

(Signat[redacted] of the G[redacted]e wi[redacted]date)
[redacted]r. Geetha V