OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
FACULTY OF COMPUTER SCIENCE

Communication and Networked Systems

Delay-Constrained Wireless Multi-Hop Networks in the Tactile Internet

Dissertation zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing), angenommen durch die Fakultät für Informatik der Otto-von-Guericke Universität Magdeburg von

M.Sc. Frank Engelhardt,
geb. am 15. April 1986 in Neuruppin

Gutachter:
Prof. Dr. Mesut Güneş, OVGU Magdeburg
Prof. Dr. Stefan Fischer, Universität zu Lübeck
Prof. Dr. Torsten Braun, Universität Bern

Magdeburg, den 26. Oktober 2022

AG Communication and Networked Systems
Institut für Intelligente Kooperierende Systeme, Otto-von-Guericke-Universität Magdeburg

Frank Engelhardt
Delay-Constrained Wireless Multi-Hop Networks in the Tactile Internet
Dissertation, Otto-von-Guericke University Magdeburg, 26. Oktober 2022.

Zusammenfassung

Das Taktile Internet verspricht weltweite haptische Kommunikation, also physische Interaktion zwischen Endpunkten, als weitere Modalität zusätzlich zu Audio- und Videosignalen. Haptische Kommunikation zeichnet sich im Wesentlichen durch hohe Latenzanforderungen im Bereich von einer Millisekunde aus (Round-Trip-Time). Weiterhin bestehen hohe Ansprüche an die Zuverlässigkeit im Sinne einer niedrigen Paketverlustrate im Bereich um 0.001 %. Der Trend wird häufig unter dem Begriff URLLC (engl. Ultra Reliable and Low-Latency Communication) zusammengefasst. URLLC ist durch physikalische Grenzen in der Signalübertragung auf eine maximale Kommunikationsreichweite von etwa 100–150 km beschränkt.

Zur Zeit lässt sich ein genereller Trend hin zu drahtloser Kommunikation beobachten. Latenzarme drahtlose Multi-Hop-Netze können somit ein Fundament für das Taktile Internet bilden. Das drahtlose Medium birgt jedoch noch zahlreiche Herausforderungen, besonders hinsichtlich unseres Verständnisses über das Latenzverhalten. Es hat sich in der Forschungsliteratur gezeigt, dass das gesamte Netz nicht einfach anhand des Verhaltens einzelner Knoten modelliert werden kann. Bisher konnte für drahtlose Multi-Hop-Netze nur der Durchsatz zufriedenstellend modelliert werden, was aber für das Taktile Internet nicht mehr ausreicht. Zudem müssen sich Applikationen im Taktilen Internet aber auch mit der Reichweitenbeschränkung von URLLC arrangieren, was noch viele Forschungsfragen aufwirft.

Mit dieser Arbeit führen wir zunächst den Begriff der *Tactile Internet Application* ein, um eine Abgrenzung zwischen reichweitenbeschränkter URLLC und potenziell weltweit ausgedehnter haptischer Kommunikation im Taktilen Internet zu schaffen. Über hohe Distanzen muss vor allem der propagation Delay mit Software-Mitteln beseitigt werden. In unserer Vision vom Taktilen Internet fungieren Digitale Zwilinge als Stellvertreter für reale Objekte, welche frei im Internet insatanziiert werden und so zumindest virtuell Distanzen verringern können.

Unser Forschungsbeitrag ist dreigeteilt und zeigt in Summe einen Ansatz, wie die Transition der bestehenden Infrastruktur des Internets hin zum Taktilen Internet gelingen kann. Zuerst definieren wir mit dem Applikations-Framework HDTF eine Softwarearchitektur, die die Herausforderungen haptischer Kommunikation über große Distanzen neu darstellt und neue Lösungswege aufzeigt. Zweitens entwickeln wir einen neuen Ansatz für ein probabilistisches Latenzmodell für drahtlose Multi-Hop-Netze, das auf einem matrix-exponentiellen (ME) Warteschlangenmodell basiert. Drittens entwickeln wir mit der Tactile Coordination Function (TCF) eine Erweiterung für den MAC-Layer in drahtlosen Netzen, das wir exemplarisch für IEEE 802.11 implementieren.

Wir evaluieren sowohl unser Latenzmodell als auch die Tactile Coordination Function mittels Simulationen und realen Testbed-Daten. Das HDTF-Framework implementieren wir zumindest teilweise in der Form des Haptic Communication Testbeds an der OVGU (OVGU-HC), als Teil des MIoT-Labs. Als Erkenntnis aus unserem Vorhaben ist eine ganzheitliche Entwicklung in alle drei Richtungen sowohl im low-layer Protokollentwurf als auch im high-level Softwareentwurf nötig, um dem Ziel einer weltweiten haptischen Kommunikation näher zu kommen.

Abstract

The Tactile Internet aims to provide worldwide Haptic Communication as the next evolutionary step of the Internet. Haptic Communication introduces a new modality of physical interaction between endpoints, in addition to classic audio and video signals. The key performance criteria are low latency in the range of a millisecond round-trip time, and high reliability in the range of 99.999 % in terms of packet delivery rate. In the networking community, the trend is picked up by the term Ultra-Reliable and Low-Latency Communication (URLLC), which is one major part of the Tactile Internet vision. However, URLLC is limited to communication distances in the range of 100–150 km. Wireless communication offers good conditions, as the speed of light for radio waves is among the highest, pushing the operating distances towards the 150 km mark.

Delay-constrained wireless multi-hop networks are thus a key enabler for the Tactile Internet, but they have effects on the application and are difficult to model and predict. The shared medium imposes a major challenge, as the behavior of the entire network can not be modeled from just the isolated behavior of the individual nodes. So far, wireless multi-hop networks could only be modeled in terms of throughput, which is not sufficient anymore for the Tactile Internet. At the current state of the art, the utility of wireless multi-hop networks for URLLC is unclear, as no protocols exist and latency can not be modeled effectively.

With this thesis, we first introduce the notion of a *Tactile Internet Application* to motivate the necessity of different approaches for short-distance and long-distance Haptic Communication. For longer distances, the propagation delay has to be overcome by means of the application software. In our vision of the Tactile Internet, Digital Twins can act as proxies of real objects, which can move and instantiate freely on the Internet. Thus, they can be a means to overcome distance-related propagation delay.

Our three-fold contribution shows how the Tactile Internet can be realized while respecting the existing Internet infrastructure and employing wireless multi-hop networks for Access Network URLLC. First, with an application framework called Haptic Digital Twin Framework (HDTF), we define a software architecture that can solve the problem of worldwide Haptic Communication. Second, our probabilistic latency and reliability model for wireless multi-hop Access Networks, based on a Matrix-Exponential (ME) queueing model, is a novel modeling technique for wireless URLLC. Finally, with the Tactile Coordination Function (TCF) for wireless MAC-Layer protocols, which we implement exemplarily for IEEE 802.11, we contribute a method for integrating Haptic Communication in existing wireless technologies.

We evaluate both our probabilistic modeling approach and the Tactile Coordination Function through simulation and real-world testbed experiments. We implement the proposed software framework, at least in parts, in the form of the Haptic Communication Testbed at the Otto-von-Guericke University of Magdeburg (OVGU-HC), a testbed extension for the MIoT-Lab at the OVGU.

We find that all three steps, from low-layer protocol development, to modeling, to a high-level software framework must be pursued together in order to achieve the goal of worldwide 1 ms-connectivity.

Dedicated to everyone who made this happen.

Contents

List of Figures

List of Tables

Listings

CHAPTER 1

Introduction

The Internet has led to a democratization of information through its capability to share and distribute information quickly all over the world. The capabilities of social and multimedia platforms so far have had a massive impact on societies, businesses, and human life in general. It is remarkable that still to this day, primarily audio and video information is shared, which has been available for broadcast for almost a century now. The Tactile Internet will change the premises so that increasing amounts of shared information will also be physical, addressing and appearing to the human sense of touch. 'Touch,' in its broad sense, covers all aspects of physical interactions, like movement, warmth conductivity, and the sensing of torque, force, and surface structure. The paradigm shift that this development initiates is known as the 'democratization of human skills' [3, 4], since now physical interaction also becomes possible through the Internet.

The COVID-19 pandemic between 2020 and 2022 has shown that a vast amount of human collaboration can be outsourced from physical presence to virtual presence, either via e-mail, chat, telephony, or video conferencing. Although this outsourcing was only possible in areas where audio and video communication was sufficient, the extent to which remote work was made possible is remarkable. The shift has been made possible through mobile telecommunications by a large part. The Tactile Internet idea is closely related to the development of the 5th Generation Mobile Communication Technology (5G) [4], which provides many key technological features [5]. Telecommunication systems have a strong influence on the Tactile Internet, as they lead to broader access to Internet services worldwide, with a strong market share, especially in non-developed countries. Bandwidth is almost not a problem anymore, and similarly, also device densities and spectral efficiency is increasing as much as applications can consume them. With the Tactile Internet, the real-time aspect comes more into focus. Both *latency* and the *reliability* of communication services are now equally important measures and are becoming core concerns for the ever more integrated digital society.

Technically speaking, the Tactile Internet will have *Haptic Communication* as its dominant payload. The shape of Haptic Communication can be determined with some detail [6] already. Haptic Applications are in use today already, although in relatively confined spaces, for example, in the form of Networked Control Systems (NCSs), Intelligent Transport Systems (ITSs), haptic displays, and telepresence systems, which are often subsumed under the term Cyber-Physical Systems (CPSs). These systems, in addition, undergo a constant development towards increasing digitalization. The ultimate goal of worldwide extension of these systems leads to the emergence of Digital Twins, which replace the interaction with distant real-world objects. The Tactile Internet can be regarded as the integration of both CPSs and the Digital Twin concept into the Internet of Things (IoT). On a low-level view, the integration is achieved by providing remote physical interactions, networked control, and the communication of touch experience [5]. Broad domains can benefit from the sense of touch, and applications range from healthcare to public services, education, and recreational use. Furthermore, not only

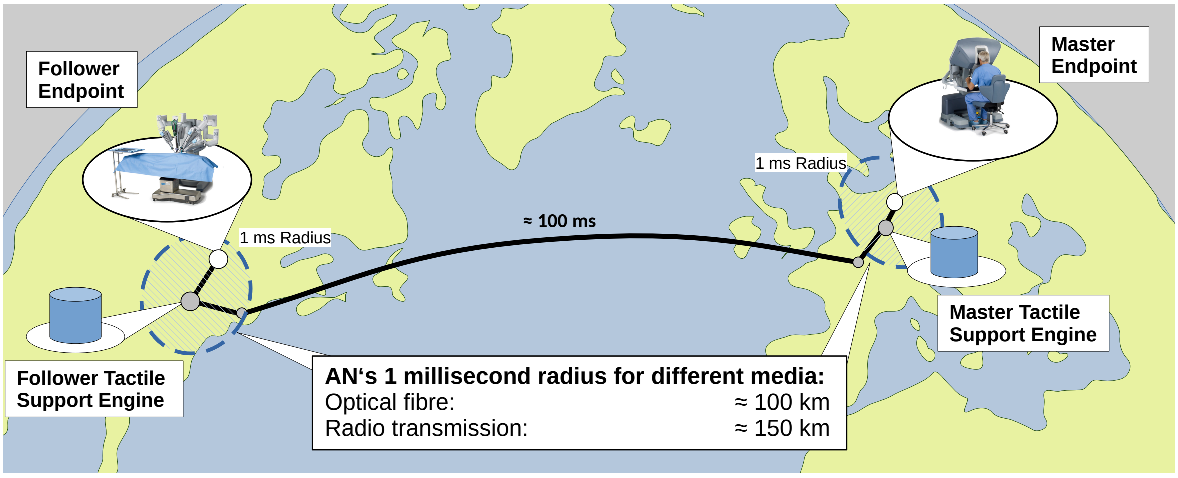

Follower Endpoint
Master Endpoint
1 ms Radius
1 ms Radius
≈ 100 ms
Follower Tactile Support Engine
Master Tactile Support Engine
AN's 1 millisecond radius for different media:
Optical fibre: ≈ 100 km
Radio transmission: ≈ 150 km

Figure 1.1: An intercontinental telesurgery scenario. Experts, like surgeons, can share their skills potentially across long distances. A perceived latency of 1 ms must be met, which as a hard requirement can only be accomplished within short distances. These boundaries typically fall together with those of Access Networks (ANs, hatched areas). Further computational effort with assisting Tactile Support Engines is necessary to extend the application boundaries.

human interaction can profit from this dimension, but also human-machine and machine-to-machine interaction.

For Haptic Communication, an end-to-end, round-trip communication latency of 1 ms, service reliability of 99.999 %, and a packet rate of about 1 kHz are often assumed. The numbers originate from the accuracy of the human sense of touch, especially the sensorimotor system, which is characterized by its very short reaction times to stimuli. It is, among other tasks, responsible for the upright stance and for reflexes, which require certain signals to be evaluated already within the spine to circumvent the slow conscious recognition within the brain. When opening the digital world to the sense of touch, technical systems must be prepared to operate in similar physical dimensions in terms of latency and reliability. These requirements are specific to the Tactile Internet, and are new categories in the realm of Internet service quality.

The paradigm of the Digital Twin is a remarkable development that is tightly connected with the Tactile Internet. Digital Twins represent the entanglement of a real object with a virtual counterpart. Accesses to either of the two parts are always reflected by the respective counterpart [7]. Driven by the digitalization of modern workflows, communication networks have become increasingly aware of this concept. Especially for the IoT, the Digital Twin is an integral part of its function. At the current technology level, the simpler Digital Shadow is often used, employing only a one-way entanglement. With future evolutions, however, the Digital Twin will mature both in terms of the breadth and depth of its representation, and become eventually a complete twin as it is incorporated in platforms and communication networks. This development ultimately allows for sharing a broader bandwidth of professional skills along the Internet.

1.1 Haptic Applications at Intercontinental Scale

As a critical concern, the Tactile Internet brings Haptic Communication to a worldwide deployment [3–5]. Clients of Haptic Applications can reside anywhere in the world, where they can cover any distance, from close-range to intercontinental scale. However, the latter is the critical ambition. So far, this distinction has never been a genuine concern for many applications on the Internet: sharing audio, video, or any other content has seldomly been restricted by the communication distance. Still, it could be saturated sufficiently by the provision of more and more bandwidth. Only a few applications today depend on communication latency; if so, they rarely also use much bandwidth simultaneously. These exceptions (financial transactions, computer games, and audio telephony) only make up a minority of present Internet traffic. Haptic Communication can be expected to have substantially higher bandwidth requirements. In this regard, the worldwide aspect poses a fundamental challenge to communication systems. The following example can illustrate this problem.

Telesurgery is one of the primary use-cases of the Tactile Internet [4, 5]. When imagining a telesurgery application on the Tactile Internet, as shown in Figure 1.1, the patient and the surgeon can be located very far away from each other, perhaps at an intercontinental distance. The Tactile Internet allows here to share the 'skill set' of the surgeon, allowing them to operate on patients without traveling. As a technical solution for the endpoints, surgical robots exist for some medical use-cases. They comprise two parts, an operator, the master domain that hosts the surgeon, and a follower, which executes the procedure on the patient. For the time being, these robots lack the support of long-ranged networked control and can only operate over very short spatial distances, usually within the same building. The communication channel should provide 1 ms of end-to-end latency and a 99.999 % reliability during the procedure to eliminate all network-related artifacts that could interrupt the operation and do eventual harm to the patient. However, as depicted in the figure, the transatlantic latency alone exceeds the required millisecond. As signals can only travel with the speed of light, physical limitations are hit for one-millisecond round-trip communication after a distance of 100–150 km, depending on the communication medium. (The author admits that surgical procedures have been conducted via the Internet already since the 1990s, e.g. [8, 9] under much less stringent network constraints, but these procedures seem to be conducted very rarely.)

As a necessary solution, the distant endpoints have to migrate certain parts of their services towards each other, as the only solution to distance-related latency can be distance reduction. This can, for example, involve some means of prediction. When the endpoints can refer to computational capacities in their vicinity, which can provide Artificial Intelligence (AI) for predicting the state of the opposite peer, the distance can be bridged with the predicted state. The surgeon, in this example, would not operate on the patient directly but on a Digital Twin of the patient, which can be a real-time procedure. The prediction horizon needs to cover the communication latency of the network between the endpoints and the AI capacity can either be provided by the endpoints themselves or by Edge Cloud applications as part of the Internet infrastructure. Using such dedicated infrastructure as a Platform-as-a-Service (PaaS) has several benefits over integrating the AI directly into the endpoints, as many processes and application parts can potentially be re-used between many applications. Additionally, the ability to migrate and scale applications can be greatly enhanced with a PaaS solution. In Figure 1.1, Edge Cloud instances at intermediate locations are called Tactile Support Engines (TSEs) according to the terminology used in literature. TSEs are part of the Tactile Internet infrastructure as envisioned by many researchers [4, 10] and by standardization organs [5].

1.2 Problem Statement

The necessary step of outsourcing the computational load to the Edge Cloud leads to a key role of the Access Networks in the Tactile Internet compared to the traditional Internet. The Access Networks must now provide the stringent *Quality of Service (QoS)* requirements of the applications (latency, reliability, and bandwidth), and only the provision of these requirements allows for the vision of the Tactile Internet to function. A core concern is the limit of 150 km to the next TSE, which is a theoretical maximum. In practice, the various communication protocols used may induce additional latency, e.g., through fixed symbol lengths, header fields, arbitration delays, or from necessary retransmissions. In addition, more latency is also added by devices like routers, Access Points, switches, bridges, and others. This *multi-hop* structure of typical Access Networks generally adds complexity to the

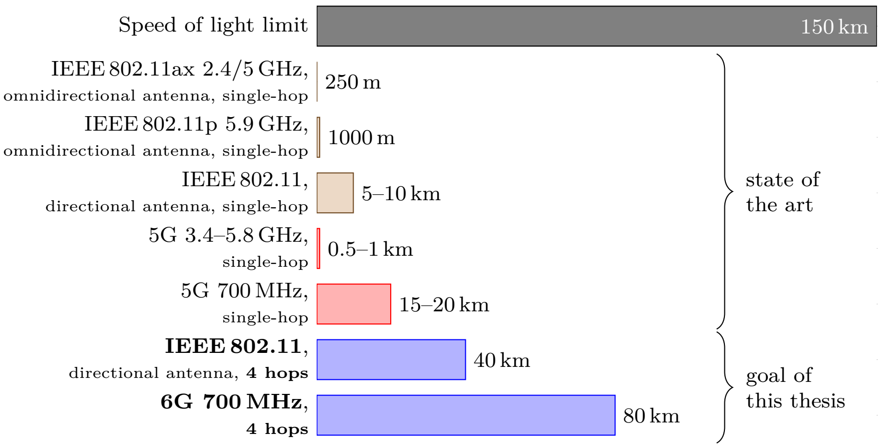


Figure 1.2: Comparison of communication ranges for Access Networks that allow sub-1 ms latency.

latency problem, but, is also clearly desired, as the theoretical range limit can be met easier with multi-hop technology. The range of single-hop links is often limited, either by cable length, by infrastructural limitations, or, in case of wireless communication, due to limited transmission power.

Wireless transmissions can be considered as the communication medium of choice for Haptic Communication, since it is capable of reaching communication peers on a direct and short path. Wired links are often longer than just the aerial distance and thus induce more propagation delay. In addition, the speed of light in air is faster than that of optical fiber or copper wires, so the theoretical limit is higher, and Access Networks can cover a larger area.

Figure 1.2 compares the theoretical maximum communication distance (speed-of-light limit for 1 ms round-trip) to that of various technical solutions for wireless Access Networks based on technologies available today. The values are estimated by the author based on link budget calculations from current hardware specifications and varying transmission power. Omnidirectional WiFi (Wireless Fidelity, IEEE 802.11) links, which are common Access Network Technology for Smart Homes and most businesses, can have ranges around 250 m due to the limited transmission power allowed within the ISM band frequencies. A modification for ITS, IEEE 802.11p, can reach up to a kilometer in range on a dedicated channel, which is still much lower than the theoretical maximum. A higher range can be achieved with directional antennas that can provide additional power gain. This can be achieved with static shaped antennas, or with beamforming by Multiple Input/Multiple Output (MIMO) antenna arrays, which would also allow for node mobility. With similar arguments, 5G also has a limited range of about the same extent, as similar frequencies are used, although with higher transmission power. A bigger difference can be made with sub-GHz frequencies, such as with the 700 MHz-band of 5G, but here only a limited number of channels and smaller bandwidths can be used. These bands also are not specified to be used with low-latency applications in 5G [11].

The example shows that a true approach to the theoretical maximum can be achieved with multi-hop communication only. In Figure 1.2, we have shown exemplary setups with 4 hops for WiFi and 5G, respectively. These solutions require wireless multi-hop networks which are capable of sustaining the required service guarantees. However, the use-case of Ultra-Reliable and Low-Latency Communication (URLLC) has not been transferred to wireless multi-hop networks in research literature, neither do solutions for URLLC multi-hop networks exist for WiFi or 5G. For this, appropriate models are required that can assess the free capacities of a multi-hop network in order to decide if a given application can be deployed.

In addition to the modeling of Access Network resources, the complexity of Tactile Internet Applications must be covered by an application framework. The endpoint migration is a central aspect to circumvent the limitation of the intercontinental communication delays.

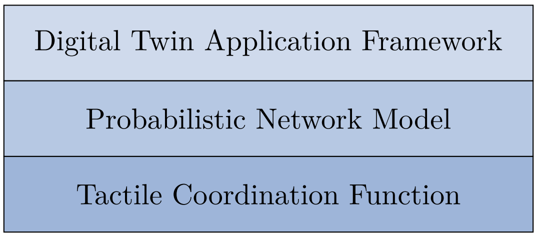

Digital Twin Application Framework
Probabilistic Network Model
Tactile Coordination Function

Figure 1.3: Thesis contribution.

As a solution, we introduce a Digital Twin approach that suits the current structure of the Internet and follows its openness aspect.

1.3 Thesis Contribution

The contribution of this thesis is threefold:

- We develop a Haptic Digital Twin Framework (HDTF) for the Tactile Internet that covers aspects of the software architecture for both Digital Twins and real hardware endpoints. This aspect is covered in Chapters 5 and 10.
- We present a probabilistic model, the Matrix-Exponential (ME) queueing model, to predict the latency behavior of wireless multi-hop networks. More precisely, we demonstrate means to model both the queueing delay on the Network Layer, as well as any delays that occur on the Host-to-Network Layer (Data Link and Physical Layers). This aspect is covered in Chapters 7 and 8.
- We introduce a Tactile Coordination Function as an amendment to IEEE 802.11 that is able to support Haptic Communication and with that, show that a multi-hop solution for Tactile Internet Access Networks based on WiFi is feasible. The solution prospects the ability to use similar multi-hop approaches for 6G Self-Organizing Networks, which are to date not a key focus of standardization. This aspect is covered in Chapter 9.

Figure 1.3 summarizes the three parts of the contribution in their hierarchical order, from top to bottom. The Tactile Coordination Function on the lower layer is responsible for the implementation of QoS for Haptic Communication in recent wireless technologies, for which we chose WiFi as an example. With that knowledge, our probabilistic model can contribute a resource assessment for the Tactile Internet, which is required to model latencies and implement resource admission and enable goal-oriented management of network resources. The framework, finally, allows to implement Tactile Internet Applications.

1.4 Structure of the Thesis

The rest of this thesis is divided into 5 parts. Part I, consisting of the two chapters Background and Methodology, introduces necessary concepts and outlines the research approach. In Part II, we focus on a concrete vision of the Tactile Internet architecture. Chapter 4 summarizes the state of the standardization efforts related to the Tactile Internet and points out open questions. In Chapter 5, we propose an open Tactile Internet architecture that is derived from the existing distributed Internet architecture. Chapter 6 begins with the basic formalization of the Tactile Internet Applications and the related parts of the network. In Part III, we specifically address the Access Networks, as they are a key component in the proposed architecture. Chapter 7 shows a latency analysis approach for WiFi multi-hop networks, which is at first omitting queueing delays stemming from protocol contention. Chapter 8 develops a separate model for estimating the queueing delay. Chapter 9 addresses the provision of QoS in multi-hop WiFi networks and introduces a protocol amendment. In Part IV, we present an application Framework for the Tactile Internet that includes Digital

Twins as a means of latency mitigation. Chapter 10 describes this framework and shows a partial implementation in the form of a Haptic Communication Testbed. Chapter 11 presents evaluation data for both the framework and the queueing model. Part V then concludes the thesis with a summary and an overview of future work.

PART I

Background and Methodology

CHAPTER 2

Background

In this chapter, we review the fundamentals required to address the problems and concepts of Haptic Communication in the Tactile Internet. First, we introduce the current Internet infrastructure, from which the Tactile Internet will evolve. We introduce the concepts of Haptic Communication and the vision behind them. Then, we introduce the terms Quality of Service (QoS) and Quality of Experience (QoE), which describe different aspects of a networked application's 'quality' dimension. We also review the most recent use-cases of Haptic Communication and their postulated requirements. Finally, we give an overview of the known quality metrics relevant specifically for the Tactile Internet and Haptic Applications.

2.1 The Internet Architecture: From ARPANET to the Tactile Internet

The Internet is a globally distributed system based on the TCP/IP network stack to distinguish internetwork communication services into five distinct layers. One fundamental advantage of the TCP/IP model is that it is not formally specified, as is the ISO/OSI reference model, which appeared later and failed to replace the TCP/IP model as the basis of the academic and engineering landscapes. The TCP/IP model is settled around the Internet Protocol (IP) family, with the initial IPv4 (RFC 791 [12]) protocol, which is now more and more replaced by the IPv6 (RFC 8200 [13]) protocol version.

The Internet developed quickly from a small-scale and particular University network to a global communication infrastructure. Originating from the ARPANET in the late 1960s, it iteratively grew more significant in a mostly unregulated and distributed manner. As often in information technology history, technologies that provide scalability and simplicity outperform technologies that offer better service quality. A prominent example is the Ethernet protocol that displaced many other protocols, such as ATM and Token Ring, which both could provide better service quality and more reliable connections. Similarly, the WiFi protocol, which provides unreliable service quality, could displace the more reliable but complex HIPERLAN technology in the 2000s. This trend shows that new technologies appearing on the Internet have to prove their simplicity and applicability during the early adoption phase, not only to engineers but also to the end customer. Therefore, the new Tactile Internet protocols and standards must also prove their simplicity and applicability in the same way. Although it is not possible to predict any success for technologies before their rollout begin, it is clear, though, that a first hurdle will be the goodness of fit towards the established infrastructure that is currently present. The current Internet is unreliable, and the Tactile Internet will have to be built on top of this architecture before anything specific will change towards a better infrastructure.

We give a brief overview of the Internet architecture here as it is. For this, we borrow the view of Kurose & Ross [14], shown in Figure 2.1.

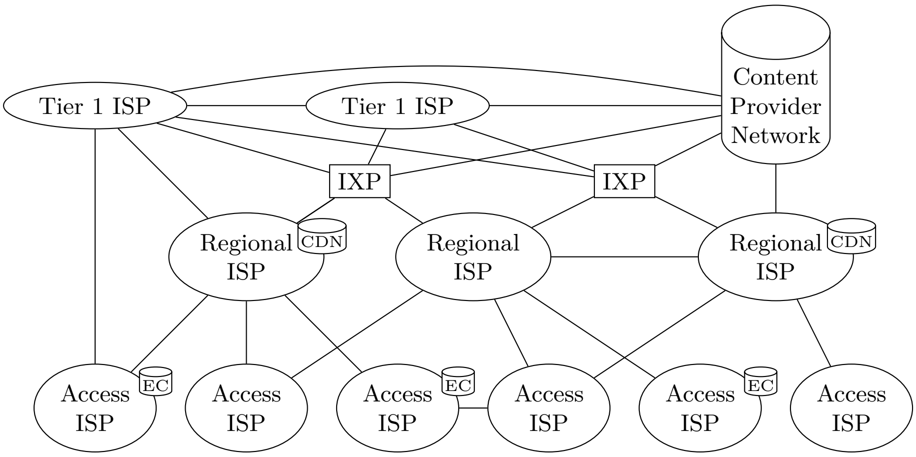


Figure 2.1: Structure of the Internet core network, adapted from [14]. Content Delivery Networks (CDNs) and Edge Cloud (EC) components are located at lower tier networks to decrease communication distances to the end systems.

2.1.1 Endpoints

When viewed as a graph, as in Figure 2.1, where computer systems are represented as nodes and interconnections between them as links, the *endpoints*, or *hosts* form the edge. Hosts run applications and implement all TCP/IP model layers. Typically, these are desktop PCs, workstations, smartphones, wearables, home or enterprise servers, things (as part of the Internet of Things), machines (robots, production machines, cars, aerial vehicles, etc.), servers in a data center (for storage, service, software, web hosting, content provision, Digital Twins, large-scale computation, etc.), sensors, actuators, smart tags, and so on. From the network's perspective, they all produce or consume data, or do both simultaneously, i.e., they can be data *sources* or *sinks*.

The Internet is organized as a tree-like structure, where the Internet edge is at the bottom and comprises the vast majority of devices. As of 2022, an estimated 30 billion devices are connected to the Internet, trending towards 50 billion in 2030 [15].

End systems can have different roles. The classic view is that information is presented by *servers* and requested, or consumed, by *clients*. The client-server communication model is often found where information has to be shared between many similar entities. Protocols like HTTP and CoAP consider this type of communication. Other forms can be *peer-to-peer*, where devices communicate in a many-to-many fashion without the need for coordination by a central entity. Another type is the broker-based (or publish-subscribe) paradigm that provides a more efficient n-to-m communication organized by a central entity.

2.1.2 Access Networks

An Access Network is responsible for connecting an end device, or a set of end devices, to the first Internet router (Edge Router) [14]. Access Networks can be wired or wireless. These networks typically solve the 'last mile' problem, where Internet Service Providers (ISPs) offer paid access to Internet services. A specific network topology is often applied where the Edge Router comprises the logical "center" of the Access Network. At the same time, all end devices are distributed around that center and linked to it, possibly with more routing infrastructure. An Access Network often covers a set of end systems within a given perimeter in space. The communication flow can be separated into the *upstream*, where end devices send information through the Edge Router to some distant end device, and into the *downstream* where end devices receive information through the Edge Router. The respective communication paths are referred to as *uplink* and *downlink*. Communication between different end devices within an Access Network can occur as well. Still, an Access Network's primary purpose is often the Internet connection through the Edge Router.

Typical wireless Access Network technologies are cellular communication networks (5G, LTE (4G), UMTS (3G), GSM/GPRS), WiFi (IEEE 802.11), LoRaWAN, and Satellite networks. Some other technologies exist which are less common or can be regarded as legacy technology, like WiMax and EDGE. Also, Bluetooth can be viewed as an Access Network technology for small-scale, short-range devices.

Access Networks can utilize wireless multi-hop topology. Here, the distinction between *Fronthaul* and *Backhaul* links can be made, depending on if a link connects to an end device or is between two relay nodes. Such topologies can be found in WiFi mesh networks and cellular networks.

Access Networks can have very different spatial dimensions. WiFi networks, for example, often have a limited range, while cellular networks can be more extensive, both in cell radius and in the number of wireless base stations. The physical extents can range from small local area networks (LAN), over campus area networks (CAN), to metropolitan area networks (MAN), from a few to ten thousand or even millions of end devices. It is, therefore, difficult to model the behavior of these networks.

2.1.3 Core Network

The core network consists of all Internet routers and is responsible for transporting all data traffic between Access Networks. It is often referred to as the *Internet Backbone*. However, for different Internet subsystems, various core networks exist. For example, mobile cellular networks build on top of their core network, so data traffic that can be routed within the network of a specific ISP may not have to leave the ISP's premises. Another example is the existence of specialized Backbones, such as the Multicast Backbone. We here do not require these distinctions and refer to the Internet Backbone as a single system of network routers that are responsible for relaying data traffic on a global scale.

Although a heterogeneous system with only a small layer of central authority, the core network relies on two operating principles: *packet switching* and the *store-and-forward principle*. Packet switching means all information is confined to time-discrete entities ("packets"). Information pieces at the end devices with arbitrary length (i.e., content such as documents, pictures, videos, holograms, etc.) will be fragmented into packets and switched individually and independent from each other to their destination. In contrast to circuit switching, resources are only occupied when actual information is exchanged, and no resource reservations have to be made. With packet switching, making service guarantees is difficult, and packets belonging to a larger piece of information can be switched over different routes.

The store-and-forward principle means that a router must receive an entire packet before forwarding it. The switching process thus requires time and for passing the switching fabrics within Internet routers. Packets arriving at an input port are stored in a piece of local memory before their destination port is determined and forwarded to that port's output queue. For each passing through a router, additional time is required for processing, storing, and forwarding a packet, which increases packet latencies.

The Internet core network is organized in a decentralized way as a network of networks [14], where different regional and global ISPs provide interconnected sub-networks. Figure 2.1 The sub-networks are organized as Autonomous Systems that appear as a network under the control of a single entity and with homogeneous routing policy [16]. Autonomous Systems can have different sizes and regional extents, ranging from regional ISPs to tier-1 ISPs with a global range. The global tier-1 ISPs cover large-scale networks that interconnect multiple smaller regional ISPs.

2.1.4 Content Delivery Networks

Content providers, such as Google, also add their own networks, which operate globally. They aim to provide data to customers with low latency at low communication distances, so their data centers are located all over the globe and connected with high-level ISPs. Lowering distances between end-users and servers is necessary since the network load is a substantial concern with multimedia streaming. Distributing content data is crucial to optimize global Internet traffic, as single data centers cannot sustain current video streaming services, for example. This trend toward content distribution has led to the ever-growing decentralization of Internet content providers. The trend goes from big cloud data and computing centers toward edge and fog computing.

Content Delivery Networks (CDNs) have been established to reduce centralized storage in huge data centers. Especially Internet video platforms, such as YouTube, Amazon, and Netflix that deliver high-bandwidth data to millions of users daily require their content to be stored and replicated within the relative vicinity of their user bases. To achieve that, CDNs operate data centers that are relatively small scale but are numerous and widely distributed down to the Access Network level [14]. These data centers often replicate (cache) raw multimedia data, while metadata and control protocols are driven by classic data centers. For CDN operators, it is beneficial to distribute these data centers massively, as they save traffic costs and increase user experience (e.g., through reduced latency and increased reliability).

2.1.5 Edge Cloud

As CDNs are often operated by specialized companies and are optimized specifically for data storage, their purpose is restricted more or less to content caching. But today, more and more services need bigger data traffic and therefore also require direct user vicinity. Edge Cloud solutions can generalize and further distribute data centers to allow for general computation. The Edge Computing paradigm has evolved from the Cloud Computing paradigm, intending to decrease the network utilization [17, 18]. Edge Cloud Servers can be placed directly within structures of Access Networks, for example, at cellular gateways or even at Base Stations. Moreover, they can host computational services and can thus be used for service replication and load distribution. A key benefit for Edge Computing services is the short communication distance to their respective end systems, increasing the latency and responsiveness.

2.1.6 Recent Developments: Towards 6G Communication Networks

Further development can be seen in the technological advancements toward system integration and service unification. The trend shows that the number of different technologies used on the Internet decreases as many providers switch to IP-based infrastructure. Another trend is the shift towards all-wireless connectivity, which kicked off with the rollout of 5G. Starting as an iterative improvement of established technologies in 4G/LTE, 5G ultimately steps towards full (perhaps worldwide) wireless connectivity coverage with seamless mobility and virtually no bandwidth limitations. 5G is the implementation of the requirements catalog IMT-2020 [11] from International Telecommunication Union (ITU) in 2017 which includes three different application scenarios: enhanced Mobile Broadband (eMBB) as an improvement of the former telecommunication services for voice, video and common data traffic, Ultra-Reliable and Low-Latency Communication (URLLC), and massive Machine-Type Communication (mMTC), which is dedicated to low-power, high device density machine-to-machine services. Table 2.1 summarizes the 5G requirements catalog.

Developments beyond 5G towards 6G will no longer only focus on further pushing bandwidth, latency, device density, or reliability capabilities. Technological integration will be addressed, too, such as the integration of satellite communication and networks carried by Unmanned Aerial Vehicles (FANETs). Also, the integration of social and ecological aspects was envisioned in early white papers for 6G [19, 20] and are still a concern in more recent literature. However, the realization of these concepts remains unclear. Latva-aho et al. [20] have published an early white paper on 6G goals and visions. Yang et al. [21] elaborated on potential technologies. Both agree that URLLC plays a big role again in 6G, and potential technologies will include Visible Light Communication (LiFi), Terahertz band radio, extended MIMO and extended mMTC capabilities. A very clear vision for 6G comes from

Table 2.1: IMT-2020 Specification [11].

Parameter	5G appl. scenario	Value
Latency	eMBB	4 ms
	URLL	1 ms
Peak data rate	eMBB Uplink	10 Gb/s
	eMBB Downlink	20 Gb/s
User experienced data rate	eMBB Uplink	50 Mb/s
	eMBB Downlink	100 Mb/s
Device density	mMTC	10^6 per km^2
Reliability (PER)	URLL	10^{-5}
Mobility	Urban eMBB	30 km/h
	Rural eMBB	500 km/h
Handoff time	eMBB, URLL	0 ms

Samsung [19]. They address four key applications for 6G: the focus on machine-type communication, support for VR/AR/XR incorporating holograms, infrastructure-based AI, and a focus on social development. These can be seen as a further manifestation of the Tactile Internet (TI) idea. The support for holographic applications and AR is seen as a key application for 6G by most researchers [22].

An explicit consent about 6G further integrates computational resources for Edge Computing, Artificial Intelligence, and Machine Learning into the network infrastructure [22]. The ultimate vision is the 'Beyond Shannon' idea, where the available bandwidth no longer determines the information shared through the Internet. Information flows are steered and enhanced by Artificial Intelligence to provide an abstract representation of information, which allows for more bandwidth efficiency. Such development goes hand in hand with the Digital Twin idea [7], which intends to entangle real-world objects with their digital representation. The Tactile Internet requires, in these terms, a 'Beyond Einstein' approach, where AI is required to eliminate the necessity of physical communication by replacing it with predicted or virtual information. Therefore, the Tactile Internet is a step in the same direction as the vision of the 6th Generation networks.

2.2 Haptic Communication

The Tactile Internet is characterized by the properties of Haptic Communication, its main traffic class. Although we have already used the term intuitively in the introduction, we now want to have a detailed view of its structure and implications. It has its origins in the *haptic modality* of the human sensorimotor system [23, 24], which denotes the human sense of touch, warmth, force, movement, and orientation. Besides the visual modality and the audio modality, it completes the set of human senses that are to appear within the digital world. Haptic Communication can aid in reaching a new dimension of immersion and fluidity in human-machine interaction. But also, machine-to-machine interaction in terms of physical information (e.g., force, torque, energy, speed, position) is very similar to the haptic modality.

In the present Internet, audio and video data streams are the current most critical QoS-sensitive classes of information. Their communication and constraints are well understood, and encoders and decoders are well developed. The haptic modality behaves uniquely, especially in terms of latency. The information processing is not connected directly to the human brain but is already processed in the spinal cord in the reflex arc. With reflexes, sensory stimulation is answered by action within milliseconds before any perception of the triggering event has arrived in the brain. The human vestibular sense thus triggers at higher rates than our audiovisual system, which is processed by the brain exclusively.

Haptic data can be one of two kinds: a) *tactile* data, and b) *kinesthetic* data represent different aspects of the human nervous system. The former stems from the sense of touch, i.e., surface composition or texture, of an object that is perceived through the human skin. The latter represents the sense of muscular movement. Both modalities have to be reflected in Haptic Communication. As they are of different origins, they are also of different shapes and thus have different requirements.

2.2.1 Applications of Haptic Communication

Applications of Haptic Communication can be found in many fields and cover human-to-human, human-to-machine, as well as machine-to-machine communication. The former two can be subsumed as Telepresence and Teleaction Systems (TPTA) [23]. The latter is generally referred as Networked Control Systems (NCSs), where data about physical values are transmitted between sensors and actuators. We will have a short investigation of these two systems.

Teleoperation and Teleaction Systems

Telepresence and Teleaction Systems allow the operation of a human operator in a distant environment [25]. They are bi-directional, point-to-point communication systems connecting a human through a Human System Interface (HSI) with a distant teleoperator. The teleoperator is a robot with dynamically controlled actuators and sensors that allow for the perception of the environment of the teleoperator. The two endpoints are connected via a network, as depicted in Figure 2.2.

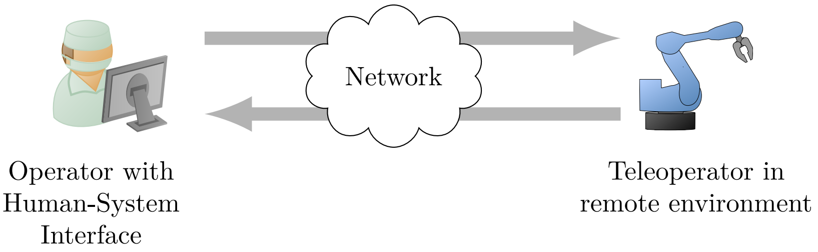

Network
Operator with Human-System Interface
Teleoperator in remote environment

Figure 2.2: Schematic of a Telepresence and Teleaction System (TPTA).

Generally speaking, there is no intelligence in the teleoperator, as it acts just as a surrogate for the human operator in a remote place. Sensors, like cameras, microphones, force/torque (within limbs and joints), touch (as in a reactive skin), or temperature sensors, perceive the environment as the operator would if they were physically present. The information flow is bi-directional. The teleoperator's sensor information must be transmitted to the operator so that the HSI can form a complete representation within both the audiovisual and the haptic domains. The operator's input then has to be transmitted to the teleoperator. For both directions, the data can have similar content, as in both ways audio, video, and haptic information can potentially be transmitted.

The audiovisual domain is nowadays very well understood, and audiovisual telepresence is common within the Internet. Augmenting TPTA Systems with the haptic domain can impose various benefits in different applications. As humans rely heavily on their sensorimotor perception, haptic feedback can improve the telesurgery and teleoperation of machines [4, 25, 26]. It is estimated that the haptic modality enables a much more complete presentation of the human skill set within the Internet, which may evolve the Internet into an 'Internet of Skills' [6, 27].

Potential evaluation criteria for TPTA Systems can be the level of immersion by which an operator can perceive the remote environment and the stability of network parameters as latency and data loss. As humans have a relatively limited and well-understood range of perception, the criteria for TPTA Systems can be derived from known boundaries of the human sensorimotor system. One of such criteria is Weber's law of just noticeable differences [23]. Weber's law states that the minimum change of a stimulus that can be detected is proportional to its magnitude. Stimuli changes of small magnitude (or energy), therefore, are unlikely to be noticeable by human perception, which allows for haptic data compression schemes similar to audio and video encoding. The thresholds of stimuli perception, such as physical pain, but also thresholds in noticeability, have been investigated quite well in the recent past [28], allowing for precise calibration of TPTA Systems.

Networked Control Systems

NCSs are formed when a control system involves communication over some kind of network [29]. Often in such cases, external sensors are connected to a control system (consisting of a *controller*, a controlled *plant*, actuators and sensors) to involve some external measurements. Examples of such systems can be external cameras for the control of a robot, the control of huge installations like HVAC systems of buildings, or water level control in a hydroelectric power plant. It depends on the application whether the network appears as part of the system that has to be modeled with its unique behavior. This is the case in most wireless communication cases, as additional delay, jitter, and packet loss must be considered when designing the controller.

Figure 2.3 shows a schematic NCS. Communication often only involves one way of communication. In most cases, controllers and actuators are built close together. In contrast, external sensors often require less energy and are not connected to the power grid.

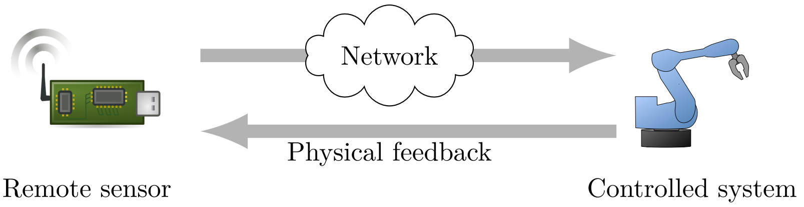


Figure 2.3: Schematic of a Networked Control System.

As the sensor information involves measurements of some physical dimension, its communication over a network can be considered as Haptic Communication. Measurements can include force, torque, temperature, velocity, acceleration, position, pressure, or multi-dimensional distance measurements (3D scans, etc.). Although the frequency can vary widely between applications, the typical range of control applications reaches up to several kHz.

For NCSs, stability is a primary performance criterion [30,31]. Instabilities occur when the control system oscillates, either because the system leaves its specified frequency range, or the network misses to deliver certain QoS criteria (mostly latency or jitter). In such cases, the control system introduces unwanted energy that might lead to malfunction and damage. A good example is given in [23], where they show that already a delay of 10 ms can destabilize a slowly oscillating system of less than 0.5 Hz frequency. Packet loss within the network can also destabilize control systems for the same reasons since losses also induce information delay. Especially burst losses are critical, as they force the controller to operate on outdated data for more extended periods.

2.2.2 Tactile Data

Tactile data represents structured, two-dimensional information regarding the surface structure of an object as perceived through the mechanoreceptors in the human skin. Therefore, the data structure must represent the pressure or temperature information intended to be presented to these receptors in the skin, or in other words, to resemble the object's surface. It is two-dimensional information since the skin's surface, or an objects' surface area is two-dimensional. Tactile data, therefore, is similar to 2D images, which are matrices of known width w and height h that hold scalar intensity values:

$$\text{img} = \left[\text{intensity}_{i,j}\right]_{h\times w} \in \mathbb{R}_{h\times w}$$

i and j are pixel indices, with $0 \leq i < h$ and $0 \leq j < w$. The intensity may carry information about either the roughness, friction, or warmth conductivity of the surface [6].

2.2.3 Kinesthetic Data

Muscle feedback of the human body's somatosensory system and orientation information from the vestibular system (located in the inner ear) are summarized as kinesthetic data. The nature of the kinesthetic modality originates in limb movement and body orientation stimuli, giving the human the sense of force and pressure in conjunction with their body movement system. It differs from the tactile modality in terms of its closed-loop character. This sense is mainly similar to a control system in the classical sense, consisting of control input data (the forward path) and feedback data. Data packets, therefore, consist of vectors representing either control input x or system response (feedback) y:

$$\begin{aligned}\mathbf{x} &= (x_0, x_1, ..., x_o)^T \in \mathbb{R}^n,\\ \mathbf{y} &= (y_0, y_1, ..., y_q)^T \in \mathbb{R}^m\end{aligned}$$

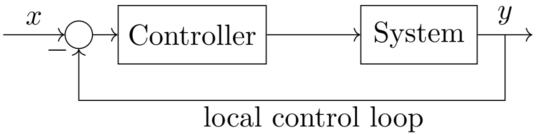


Figure 2.4: Control loop notation.

Both control and feedback may consist of velocity, position, acceleration, or force information.

2.2.4 Performance Requirements for Haptic Communication

As we have elaborated, two performance criteria can be subsumed as the core of Haptic Communications: latency and reliability. This is also a common conclusion in literature [4, 6, 10, 32], and has been recently the focus of modern telecommunication systems, starting from 5G with the IMT-2020 specification introduced in the previous section. With the upcoming of the Tactile Internet, latency and reliability are gaining more and more attention by Industry and users. Moreover, the latency and reliability requirements of Tactile Internet Applications will be more stringent than for audiovisual data transmission.

Consequently, the Tactile Internet must focus on the *Quality of Service (QoS)* aspects within its whole infrastructure. The central challenge is often summarized as "1 ms-Challenge, i.e., achieving a round-trip latency of 1 ms at an outage of about 1 ms per day" [24].

2.3 Quality of Service

In this section, we formalize the three basic QoS properties that are relevant for the Tactile Internet. These are *latency*, *reliability*, and *throughput*. The term Quality of Service developed over time as a concern on the Internet, as different services emerged that require different non-functional properties for their communication. For example, Voice over IP (VoIP) requires low jitter at a constant throughput to function correctly, while the classic download of documents does not have any specific requirement. Many approaches to establishing QoS on the Internet have emerged over time. For IP-based communication, the *DiffServ* [33] approach has been accepted as means for handling QoS constraints given by applications. Instead of explicit resource reservations, a QoS class is assigned to each packet so that the Internet routers can differentiate and prioritize high-QoS packets over low QoS packets in a per-hop behavior. The class membership is implemented by labeling with specific 8-Bit fields in the IPv4 and IPv6 headers, respectively.

The implementation of QoS handling on the Internet has developed as a set of best practices since the labeling is based on a commonly defined classification of service types. Such service types will also emerge for the Tactile Internet in the future. But currently, the range of possible Tactile Internet Applications is not yet clear. Therefore, for this thesis, we focus on QoS properties from the application's perspective and treat them as quantities that can be required by the application and possibly reserved and then granted by the network. Such a strategy is known as *Integrated Service (IntServ)* [34, 35], which has been anticipated but could finally not establish in the decentralized architecture of the Internet.

2.3.1 Latency

Latency is defined as the time delay occurring between the generation of an event and its processing. In communication systems, the events have classically generated a message from an application on a source host in the network. The processing, in this case, is completed when the message reaches the application layer on its destination host. Communication systems always introduce delays into applications, as the process of transmission always requires time. The amount of time necessary varies based on many factors, for example, channel bandwidth (or, more abstractly, channel throughput), communication distance, processing speeds in the transmitter/receiver, queueing delays, or protocol-specific mechanics such as arbitration delays or inter-frame-spaces. Latency is often used interchangeably with the term *delay*, which we also use equivalently throughout this thesis.

The effect of latency in Haptic Communication systems is a loss of transparency through delayed execution of user commands. As in Haptic Applications a control loop is closed over both ends of the communication, and latency has a big destabilizing effect in such systems [23]. In virtual environments, latency also leads to motion sickness [36, 37]. In control systems, latency has to be carefully modeled within the system design to circumvent oscillation and other destabilizing effects. Communication latency, therefore, reduces the bandwidth of the controller. Methods for circumventing the effects of latency in Haptic Applications include *passivity-based control* and *model-mediated* teleoperation [38]. However, even with such methods, reducing communication latency remains a key challenge for Haptic Communication.

In networked systems, the *end-to-end (e2e) delay* is one of the most commonly measured parameters. It is the delay in sending packets from a source to a destination host, i.e., from one Application Layer entity to another. This value abstracts from the networking infrastructure between the two hosts, regardless if it is a single LAN connection between them, if both are part of an enterprise network, or if both communicate over the Internet at an intercontinental distance. The e2e delay can be measured directly by transmitting packets from source to destination, and comparing the timestamps of their transmission at the source and the reception at the destination, given that both clocks are synchronized. When clocks are not synchronized, the Round-Trip-Time (RTT) can be measured as an alternative. Here, the destination sends back an acknowledgment message to the source so that the source can measure both the sending and the receipt timestamps with the same clock. For an estimate of the e2e delay, the RTT can be divided by 2, assuming that the communication channel is symmetric. As the latency is often not time-invariant, a measurement can only give a constrained image of the true behavior of a communication system. More accurate prediction methods are thus often desired, together with an uncertainty estimate.

Instead of the e2e delay, we often consider the *path delay* $t(P)$ on a certain path P in a network, which includes a multi-hop communication between two given nodes, which do not necessarily need to be hosts. The path delay is otherwise equivalent to the e2e delay when the path P includes the source and destination hosts of a flow on the Application Layer at both its ends. This distinction is necessary as we will later focus on path delays within Access Networks only and leave out the Backbone part of the e2e latency.

Latency must always be greater than zero and behaves additively, i.e., it originates part by part from different sources that sum up to yield the entire path delay $t(P)$. These parts can be classified into the following categories.

- The *contention delay* $t_{\text{cont}}(v)$ refers to the delay that stems from the contention of the nodes for medium access. In networks with limited communication channels, like Wi-Fi, the nodes need to order their transmissions collision-free. Since the medium is a shared resource, this delay depends on the network load. It does, however, not depend on node processing delay.

- The *arbitration delay* $t_{\text{arb}}(v)$ is required for Random Access Protocols for medium arbitration. The Carrier Sense Multiple Access (CSMA) protocol in WiFi, for example, schedules medium access based on waiting times of nodes. With channel partitioning or taking-turns protocols, this delay is zero.

- The *transmission delay* t_{trans} is the time the actual transmission takes to put a packet onto the medium. This time depends on the packet size and the bitrate of the transmitter.

- The *propagation delay* $t_{\text{prop}}(v, u)$ is the time the signal needs to travel between sender and receiver nodes. It depends on the Euclidean distance $d(v, u)$ between the nodes v, u and on the speed of light c_M in the medium M:

  $$t_{\text{prop}}(v, u) = \frac{d(v, u)}{c_M}$$

  The propagation delay can often be neglected for small-scale networks or delay-tolerant communication. However, long communication distances of several kilometers may produce non-negligible propagation delays for the Tactile Internet. We will focus more on the propagation delay in the following subsection.

- The *acknowledgment delay* t_{ack} is the delay that is required by the receiver to signal transmission success or failure. Not every communication protocol includes the transmission of acknowledgment messages, but when they are used, their handling requires additional time and often blocks the medium from productive payload transmission.

- The *queueing delay* $t_{\text{queue}}(v)$ occurs when a packet has to wait in some buffer on a node before it is processed. Queueing delay can occur on any intermediate node in the network and is present due to the customer-server problem. We also subsume any processing time for forwarding under this term.

We denote the parts above as functions of nodes v and u, respectively, that are involved. The delays are subject to the protocols that are in use, so the node as an entity in protocol execution affects its outcome. In addition to that, the given delays are also time-varying,

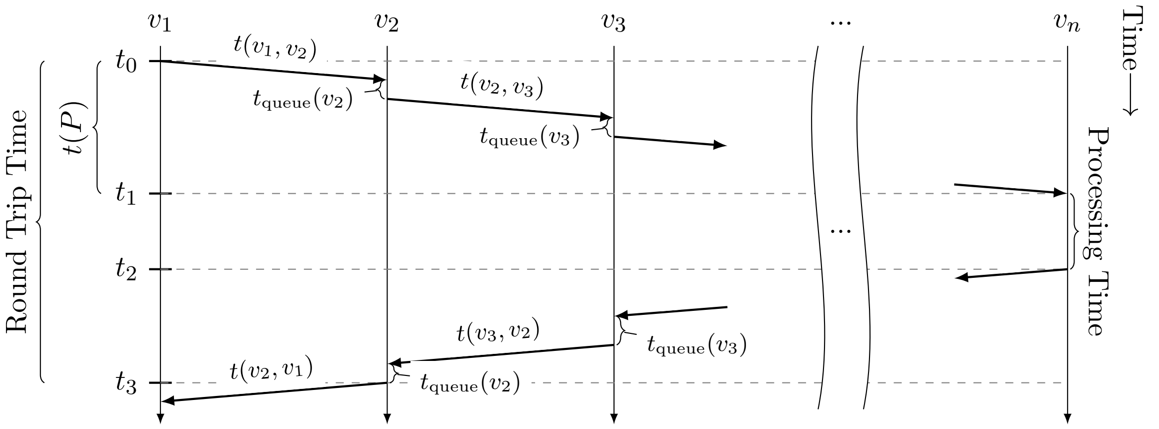


Figure 2.5: Visualization of the origins of communication latency. A packet is sent from node v_1 to v_n (the path at the top), and v_n then acknowledges the receipt (backward path on the bottom). The time dimension is plotted from top to bottom. The path delay $t(P)$ constitutes of transmissions, represented by $t(\cdot, \cdot)$ and queueing delays $t_{\text{queue}}(\cdot)$. For the RTT, additionally the processing time of the destination v_n must be considered.

Table 2.2: Signal propagation speeds in different media.

Medium	Signal speed c_M	Round-trip dist. traveled in 1 ms
Cat5 cable (twisted-pair copper)	191 867 173 m/s	96 km
Optical fibre	203 944 609 m/s	102 km
Radio waves in air	299 704 944 m/s	150 km

meaning that, for example, the contention delay at a node can differ widely between consecutive transmissions on the same node. Due to readability reasons, we omit a general index t for time throughout our further analysis, but strictly speaking, the current event time is an additional parameter of these functions. As a simplification, we summarize the contention, arbitration, transmission, propagation, and acknowledgment delays as the *single-hop delay* $t(u, v)$ between two nodes u and v, as they are subject to the transmission process between two neighboring nodes.

Figure 2.5 shows a visual representation of the delay problem in a multi-hop networked system, where a packet is transmitted over a multi-hop path to a destination node v_n. The vertical time axis indicates that time is consumed several times in the process for different reasons. Depicted here are the queueing delays and the single-hop delays. All of these functions can be addressed for latency minimization. This often has to happen on the MAC-protocol level, which is responsible for most of the mechanics that cause latency. But also the reduction of queueing delay and increased bandwidth through the physical layer can reduce latency.

The Role of Propagation Delay in the Tactile Internet

From all the potential latency sources, the propagation delay is the quantity that first comes to its physical boundaries when worldwide communication is desired. There is some variability based on the communication medium, as the speed of light varies between different materials. Table 2.2 gives an overview. Given the speed of light c_M as approx. 2.998×10^8 m/s in air, the 1 ms RTT bound restricts the distance between two communication peers to a maximum of 150 km with pure wireless communication. This boundary is already reached at about 100 km with optical transmission through glass fibers. The communication between two points on the opposite sites of the globe, which would span a straight line distance of about 20.000 km, would result in a round-trip propagation delay of 200 ms, assuming a fiber optical connection. Notably, radio transmission offers one of the highest potentials here to overcome long propagation delays, so it is the favorable choice for low-latency communication.

In many small-scale communication networks, like WiFi networks or within the air interface of cellular networks, the propagation delay is often of relatively small influence, as the distances are short. Here, it often only affects the slot times or symbol lengths of the Layer-2 protocols, where symbol lengths must be chosen to include the propagation delay at the desired communication distances. In the WiFi protocol, for example, the slot time varies around 10 µs, which is enough to cover the desired few hundred meters of communication range [39]. Apart from that circumstance, propagation delay often only becomes a design challenge in very specialized applications, like space missions [40]. Here, the propagation delays range from 7 ms to low-earth-orbit, 120 ms to geostationary orbit, 1.3 s to the Moon, between 3 to 21 min to Mars (depending on the current distance between Earth and Mars) and up to 5 h to space probes like Voyager 1 and 2.

The propagation delay is essential for long-range communication. The ratios can be visualized when we consider, e.g., a 4K video streaming application between Berlin and Sydney, where we estimate some of the involved sources of latency in the following Figure 2.6. The portions of time needed for several communication steps are typically well-known and can be measured (see, for example, Bachuber et al. [41] for an in-depth analysis of delay for real-time telepresence systems). The example shows that especially the influence of the Access Network is small compared to an estimated propagation delay of 280 ms, which we derived from `ping` measurements.

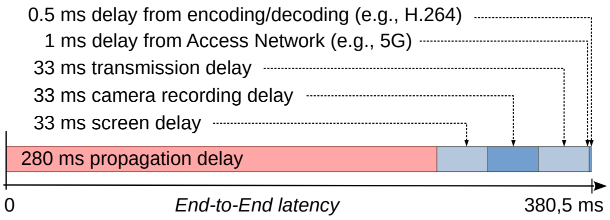


Figure 2.6: Decomposition of the communication delay for a hypothetical video-streaming example between Berlin and Sydney.

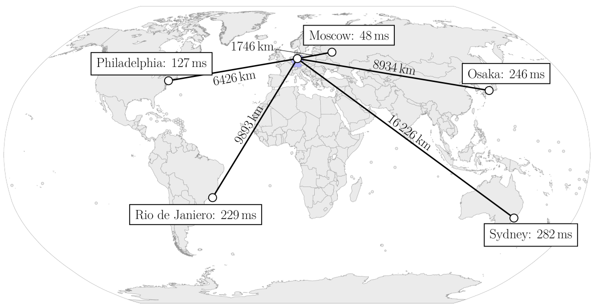
It is also notable that nearly the entire e2e delay is caused by propagation delay. We want to show that on an intercontinental scale, a delay measurement with the tool `ping` is often a good estimator for the propagation delay, in contrast to, e.g., the hop count. Figure 2.7 illustrates this hypothesis on some RTT measurements on the Internet conducted in mid 2022. Here we compare the measured RTT with both the straight-line distance and the hop count. As can be seen, the RTT shows a linear relation to the distance, while for the hop count, there is only a small correlation, as seen from the regression line. This is surprising, as on the Internet, as a store-and-forward network, the various transmission and queueing delays seem only to have a small effect on the RTT measurement.

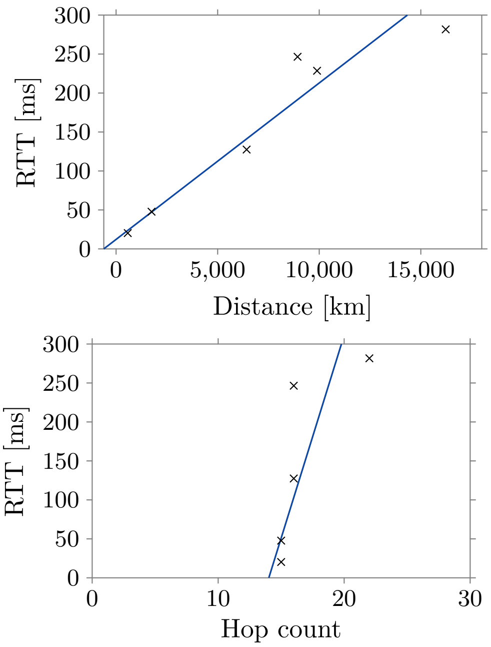
We conclude from the experiment that for the Tactile Internet, not the communication delays within the Access Networks but rather the overcoming of the propagation delay is a major challenge. It surprises in a way that in most literature, the investigation of Access Networks, especially 5G and 6G, seems to be over-represented. The propagation delay has to be dealt with in Software, by means of prediction and AI, as is more or less consensus in literature [4, 5, 24, 38, 42, 43]. Therefore, a software framework must set a foundation for such a solution.

2.3.2 Reliability and Availability

Reliability and availability are two attributes of the dependability of a system. In the literature, both terms are defined as follows [44]:

Reliability is a measure of the continuous delivery of correct service. Equivalently, it is a measure of the time to failure.





Destination	Air distance	Hops	Mean RTT
Philadelphia	6426 km	16	127.44 ± 1.57 ms
Moscow	1746 km	15	47.72 ± 0.02 ms
Osaka	8934 km	16	246.49 ± 0.15 ms
Sydney	16 226 km	22	281.66 ± 0.23 ms
Rio de Janiero	9893 km	>30	228.56 ± 5.81 ms
Zürich	574 km	15	20.28 ± 0.02 ms

Figure 2.7: A ping experiment conducted in April 2022 showing measured RTTs from Magdeburg, Germany, to servers in cities around the world ($\alpha = 0.01$, $n = 1000$). The scatter plots on the right show a strong correlation of the RTT with the distance, while the hop count has a much smaller correlation. Regression lines are plotted in blue.

Availability is a measure of the delivery of correct service concerning the ability of repairs. Equivalently, it measures the portion that a system is up over its whole service time.

Reliability defines the ability of a system to operate for a given time without interruption. For communication networks, both terms are often used interchangeably as the probability that a sent packet will arrive at the receiver without error [45]. For control systems, the distinction is, however, necessary, as the outage of a packet might result in the outage of the whole control system in the absence of redundancy. It is here necessary to review the distinction of the three dependability impairments *fault*, *error*, and *failure* [44]. A failure of a system or service means that a system fails to deliver its service correctly as defined by its specification. A failure results from a system state that is considered an error. Not every error state must lead to a failure when the system is designed to handle the error state. A fault is the cause of an error.

It is often more appropriate to handle the dependability of a communication system in terms of availability since it must maintain its service continuously. In the IMT-2020 specification, for example, the availability requirement for 5G is given in terms of the *Packet Error Rate (PER)*, which is a measure of availability. The PER is given as

$$p_{\text{per}} = 1 - \Pr\left\{\text{A sent packet is received without error}\right\}.$$

The PER for 5G is required to be $10^{-5} = 99.999\,\%$, which in terms of availability corresponds to an outage of 315.36 s in a year.

In general, the availability of a system can be increased by redundancy. For communication systems, additional resources can be allocated for fault-tolerance of transmission, like additional bandwidth (e.g., by appending checksums to the payload), additional transmission time (e.g., by allowing retransmissions), transmission on multiple channels, or transmission over multiple paths in a multi-hop network. The fundamental idea is the k-out-of-n redundancy ($k$oo$n$), where k out of a total n possible systems states are required for correct operation.

Table 2.3: Predicted QoS requirements for future Tactile Internet Applications within the kinesthetic modality.

Application	Latency	Reliability	Msg. rate	DoF	Msg. size
Augmented Reality (AR)					
Head pose only, 6 DoF	1 ms	99 %	100 Hz	6	24 B
With haptic glove, 6 DoF + 2 DoF per finger	1 ms	99 %	100 Hz	22	88 B
Teleoperation/telesurgery					
Typ. 2 end eff., each 6 DoF [46,47]	1 ms	99.99 %	1 kHz	6-12	24–48 B
Process automation					
General [48]	1–50 ms	$1 - 10^{-9}$	1 kHz	var.	10–300 B
Manufacturing process [48]	10 ms	$1 - 10^{-6}$	1 kHz	var.	<50 B
Automated Guided Vehicles (AGVs) [48]	10–50 ms	$1 - 10^{-9}$	1 kHz	3	<300 B
Networked Control Systems (NCSs)					
Inverse pendulum [30]	40 ms	99.9 %	50 Hz	1	4 B
Intelligent Transportation Systems (ITS)					
Sensor data, 3 DoF per sensor	10–100 ms	99.9 %	10–100 Hz	<100×3	<1200 B
Platooning (vehicle pose) [49]	10 ms	99.9 %	100 Hz	6	24 B
Infrastructure backhaul [50]	30 ms	99.9999 %	var.	var.	var.

2.3.3 Throughput

The *throughput* within a communication network is the amount of data that a node can send with its network resources. It depends on the noise and interference at a given time but also on the channel utilization and the processing speed of the node. The channel capacity can be derived from the physical properties, which we exemplify in the Appendix for some transmission technologies. For many wireless technologies, it is not a fixed value but depends on factors such as the used (physical) bandwidth, the frequency, the channel coding, and the noise level. The term *goodput* is often used as the corresponding measure on the Application Layer, which reduces the throughput by the amount of overhead introduced by communication protocols of all lower layers. Applications require a certain *load* from a network. The load is an application requirement, while goodput is the corresponding offer from the network side. The load calculates as the *message rate*, that an application requires, times the *message size*.

2.3.4 Typical Requirements of Tactile Internet Applications

As Tactile Internet has not yet become a reality, quantifying the requirements of QoS is difficult. Many of such predictions exist in the literature, though, which are estimates from experts in the respective application fields. One big survey of such requirements can be found in Holland et al. [5], which includes audio, video, and haptic modalities. Other surveys have been published by Jiang et al. [51], Promwongsa et al. [52], and Sharma et al. [53].

For the quantification of the kinesthetic modality, we have calculated some of the QoS constraints in Table 2.3. Our calculations are based on the description of specific Haptic Applications in the literature, where we have extracted the amount and the dimension of sensors used in the systems. The Degrees of Freedom (DoF) of an application can be derived from the number of sensors involved times the DoF per sensor. As an estimate for message sizes, we multiply the DoF by the size of an IEEE 754 single-precision floating-point number, which is 4 B. Message sizes are to be interpreted as Application-Layer quantities, not including header information of lower layers. Note that using this methodology, we conclude slightly different numbers for some applications than Holland et al. [5], which is subject to the novelty of the research field and the absence of long-term, in-field studies on such applications.

2.4 Quality of Experience

The Quality of Service metrics are properties from the network domain and are thus often too abstract for applications. They have to measure their performance in unique metrics, which can sometimes be translated into the domain of QoS metrics. But often, this translation is ambiguous or at least not trivial.

For example, in a video streaming application, a classic metric would be the video resolution (e.g., 4K or 8K) and a frame rate (e.g., 60 Hz). These properties translate to a certain throughput requirement on the QoS domain: we need to multiply the image size in pixels by the color depth and then again by the number of transmitted images per second to get the required Bytes per second of throughput.

The user, meanwhile, does not care about such calculations but is merely interested in seamless video quality. At this moment, it is unknown what the term *seamless* means in this context – even to the user. For example, it might be a valid option for a video streaming server to temporarily reduce the image resolution during a traffic spike to compensate for increased load. The user will notice the loss of image quality but is, in exchange, able to continue viewing without interruption. The dimension of 'seamlessness' is a complex metric from the Quality of Experience (QoE) domain, which is more oriented toward the user interface than the rather abstract "throughput" metric from the QoS domain.

A recent definition for the Quality of Experience has been formulated by the Qualinet Consortium in 2013 [54]: "The Quality of Experience (QoE) is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user's personality and current state." This definition regards the QoE as a quality aspect of the user experience. Especially the above example of trading image quality with the perceived continuity of a data stream is well represented here, as it results from the user perception, but not necessarily from the QoS point of view. We see that the QoE definition can leave some degrees of freedom in terms of their fulfillment, especially when seamless perception is regarded as the main aspect.

2.4.1 QoE Metrics for Haptic Communication

As QoE metrics are unique to the specific application, we have to focus on those that are relevant to Haptic Communication. Many have already been specified in recent literature, from which we want to give an overview here. Steinbach et al. provided a deep discussion on QoE metrics for Haptic Communication in [6]. Metrics can be distinguished between *objective* (i.e., metric) and *subjective* measures. Objective measures are calculated by certain rules from the affected data sets and thus yield deterministic results. Subjective measures depend on ratings from users, such as using questionnaires like the net promoter score [55].

Objctive Metrics

Some basic and general objective QoE metrics are the Mean Square Error (MSE) or Rooted Mean Square Error (RMSE) metrics. They are defined as a distance function between some original signal x_i and its transmitted counterpart $\hat{x}_i$ as follows.

$$\begin{aligned} e_{\mathrm{MSE}} &= \frac{1}{N}\sum_{i=1}^{N}\left[x_i - \hat{x}_i\right]^2 \\ e_{\mathrm{RMSE}} &= \sqrt{e_{\mathrm{MSE}}} \end{aligned}$$

Objective metrics are applied especially in control systems, where they measure, for example, the correctness of a generated control signal compared to the desired system output. They penalize even small differences and outliers in signal fidelity through the squared difference term. They are suitable for applications in Industry, Robotics, and real-time systems. Both

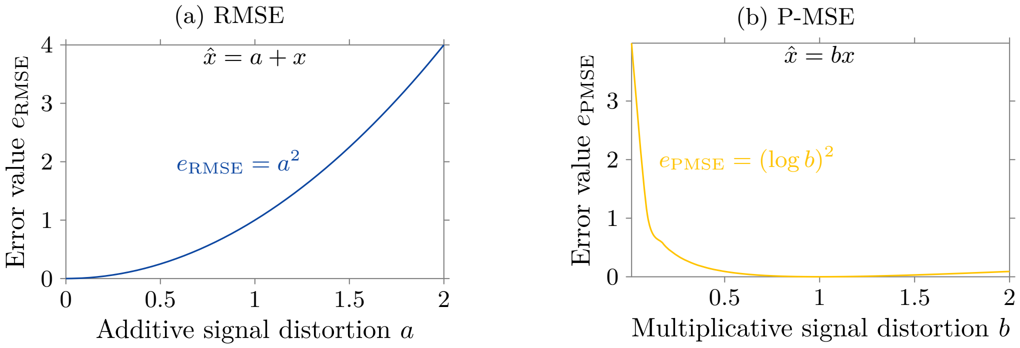


Figure 2.8: Response of the error values RMSE and P-MSE to their specific distortion types: RMSE responses to additive distortion a with an error of a^2, P-MSE responses to multiplicative error b with $(\log b)^2$.

metrics can often be used interchangeably, especially for optimization problems. They differ only in the fact that the RMSE expresses the error in the same physical dimension as the signal values reside in, while the MSE value is of the squared dimension.

The human sense of touch behaves non-linear in terms of its responsiveness to changes in value, so the MSE and RMSE are often not suitable for human-to-human or human-to-machine interaction. This psychophysical aspect is reflected by the Perceptual Mean Square Error (P-MSE) [56], which is calculated over the square of logarithmic differences between the signals:

$$e_{\text{P-MSE}} = \frac{1}{N}\sum_{i=1}^{N}(\log x_i - \log \hat{x}_i)^2$$

The RMSE is sensitive to additive distortion, while the P-MSE is sensitive to multiplicative distortion. Let assume that from the original signal $\mathbf{x} = x_1, x_2, ..., x_n$, a single arbitrary data point x_i is distorted by either an additional constant a ($\hat{x}_i = x_i + a$) or by a multiplicative constant b ($\hat{x}_i = x_i \times b$). For P-MSE, the resulting error is $(\log b)^2$, while for RMSE it is a^2. The resulting plots of these two functions are shown in Figure 2.8. It shows that the P-MSE responds mainly to signal damping, where the error penalty increases as the output signal $\hat{x}$ loses energy. We can interpret this metric as an optimization against extensive signal dampening in haptic teleoperation, where it is often beneficial for a haptic codec to reduce the signal energy to increase the overall system stability [23,38]. Dampening occurs in control systems with increasing latency. However, as it penalizes signal amplification much less, the P-MSE metric is unsuitable for control applications.

We want to mention another objective signal-level measure here for completeness. The HSSIM metric, proposed by Hassen and Steinbach [57], is defined as

$$e_{\text{HSSIM}} = \frac{1}{N}\sum_{i=1}^{N} S_p(x_i, \hat{x}_i)^r,$$

with $S_p(x_i, \hat{x}_i)$ representing a quantity measure respecting content-related signal differences in luminance, contrast, and structure, and r is a scaling power. The function provides a decomposition of the signal in these factors that also considers temporal relations. A full definition is given in [57]. We find, however, that both e_{HSSIM} and $e_{\text{P-MSE}}$ often yield similar results, so we will not go into further detail about HSSIM.

Metrics from other domains can also act as objective metrics. Verburg et al. [58] define quantitative metrics for effective time and value offsets. These metrics can assess sample-wise misalignment. It is especially interesting when omissions and/or duplication occur so that an input sample at time i does not correspond with the output sample at time i. Polachan et al. [59] define a Quality of Control (QoC) metric based on controller step response analysis.

It describes the difference between the actual control system and an ideal controller. We, however, do not go into detail about these metrics here.

Subjective Metrics

A measure that takes also the subjective user experience into account is proposed by Hamam et al. [60] for Haptic Audio Visual Environments (HAVEs). It is defined as a linear blend between some QoS metric e_{QoS} and some user experience metric e_{UX}:

$$e_{\text{HAVE}} = \zeta \times e_{\text{QoS}} + (1 - \zeta) \times e_{\text{UX}}.$$

In [60], Hamam et al. define questionnaires and derive the mathematical methods to evaluate the e_{UX} term. This metric is capable of assessing a haptic application on the system level, in contrast to the aforementioned metrics, that only cover the signal level. However, adjusting such linear blended metrics is prone to be arbitrary.

Another measure of subjective user experience is questionnaires. Questionnaire design is complex, and many aspects must be considered to eliminate biases and systematic errors. The statistical interpretation can also be difficult and involves special care and a certain professional level. Since the tools and experts may not be available, or the time and monetary budget may not allow for questionnaires, more trivial systems are often used. They also have a scientific basis but provide less information, such as common school grades or star ratings known from many social networks or from Internet shopping. Another measure with a strong scientific background is the net promoter score [55], which reduces to just one question: “Would you recommend this system to a friend?” Although this information is of rather limited value for engineers, it is trivial enough that a high percentage of system users can quickly be mobilized to give feedback. That provides strong evidence for overall system applicability. Moreover, it allows for comparison to alternative products of similar functions.

In addition to the aforementioned, many other qualitative and quantitative metrics can act as a subjective metric. Blumenthal et al. [61] have demonstrated that the *level of detail* can be an interesting metric for virtual environments, e.g., for robotics. They have developed an approach to transmit Octrees over lossy wireless networks, such as LTE, where they implemented a tradeoff between information timeliness and information resolution. The user-specified level of detail is used as a means to decide whether more detailed information should be transmitted or if a transmission should stop and other data get priority.

2.5 Medium Access Schemes for Wireless Access Networks

The end-to-end latency depends mostly on the lower communication layers, the Physical Layer and the Data Link Layer, as they determine the medium capacity and the medium access schemes. The medium access schemes are responsible for the scheduling policy in which nodes access the medium.

Medium access schemes can be divided into *channel partitioning*, *random access*, and *taking turns* protocols [14], where only the former two have influence nowadays. Channel partitioning protocols divide the channel into slices that are allocated exclusively to individual nodes for a given time period. They are collision-free, as a central instance coordinates the allocation. On the other hand, random access protocols are not centrally coordinated and use a randomized process to resolve mutually exclusive medium usage.

Channel partitioning schemes have been dominant in telecommunication systems, like 5G, while for technologies like WiFi, that operate in a free frequency spectrum, random access protocols have been dominant.

2.5.1 Channel Partitioning Schemes

Channel partitioning schemes in telecommunication networks divide the channel into frames of a given duration, which are further divided into time slots. The division can cover multiple channels with multiple bandwidths. The set of available channels and time slots within a cell is the set of channel resources available to that cell, which the individual nodes can allocate. Resources can be allocated either to the uplink or downlink portion.

The simplest scheme is the *fixed Time Division Multiple Access (TDMA)* scheme, where the size of each time slot is fixed, and each node is assigned a specific time slot for communication. A node can drop the time slot if no packets are to be sent, but no means exist to share channel resources between nodes in case of asymmetric load distributions.

More sophisticated schemes involve reservation phases within a frame to enable load balancing when asymmetric loads occur. For example, with *reservation TDMA* [62], a fixed number of k time slots can be reserved within a reservation time slot, where nodes with high demand can reserve more than one time slot, while nodes with no communication load are not required to reserve any resources at all. These and similar schemes are used in 4G (LTE) and in 5G. The advantage of such schemes is that the channel resources can be utilized very efficiently while the overhead for reservations is low. Collisions cannot occur as long as the channel resources (frequencies and medium time) are exclusively available to the communication system and no external interference exists. As a drawback, the strict enforcement of such protocols requires effort in terms of computational power and hardware for accurate time synchronization. The exclusivity required for the channel resources makes channel partitioning schemes feasible frequencies under the management of governmental authorities and thus unsuitable for channels within the ISM bands.

2.5.2 Random Access Schemes

Within the ISM bands, random access schemes are dominant due to their lack of central management. As in the ISM bands no regulations exist on the communication protocols used, collisions can not be excluded by design, and nodes have to avoid and resolve collisions if possible.

The simplest random access protocol is *ALOHA*, where no coordination happens and nodes just send when their transmission buffer is occupied. Collisions can be detected by a decoding or checksum error, which can then trigger retransmission. CSMA protocols involve a carrier sensing before transmission and defer the transmission if the channel is sensed occupied by another carrier. If blocked, the sender waits a random amount of time until the next attempt is made.

IEEE 802.11 uses a similar CSMA with Collision Avoidance (CSMA/CA) scheme where prioritization is possible due to different minimum waiting times, called Inter-Frame Spaces (IFSs). The prioritization, acknowledgments, and coordination frames from Access Points (APs) can receive higher priority, as they are used to ensure minimal but efficient coordination of network resources.

CHAPTER 3

Methodology

In this chapter, we describe the methodology used in this thesis. We outline the scientific methods and the experimentation facilities and also introduce some basic formal notation.

3.1 Empirical vs. Model-Driven Argumentation

We want to discuss and give a rationale for the model-driven approach that we pursue in this thesis. Scientific knowledge can be gained in various forms. Two kinds of argumentation, empirical argumentation and model-driven argumentation, are prominent and widespread and often compete with each other, especially in the field of computer science. The former is concerned with observing facts of the real world, carefully classifying and measuring relations between entities that exist in the present, and then abstracting rules and hypotheses from these observations. The latter way of argumentation is concerned with proper modeling of relations by laying down clear axioms and then using logical methods to derive theorems and gain new insight of higher-order from those axioms. Both approaches have advantages and disadvantages, and different “schools” exist within computer science that credit one type of approach over the other. However, it is safe to argue that many scientific questions prefer to be addressed with either of the two approaches, while the other approach produces only marginal results.

In the field of computer systems performance analysis, both of the approaches can be taken. Empirical argumentation often relies on measurements of real-world systems to predict their behavior in different scenarios. In contrast, analytical methods rely on modeling specific scenarios using, e.g., queueing theory, Markov modeling, Petri Nets, or any other modeling tool.

In the specific field of this thesis, the analysis of latency in wireless multi-hop networks however, analytical approaches tend to be more difficult, and the real-world behavior seems to escape the analytically derived predictions very often. Therefore, measurements are the preferred way to argue about the performance of a given network, and models of networks often are abstractions of empirical data of similar networks. Empirical data is herein often expressed as a set of model parameters (such as processing speeds, expected service rates of queues, expected throughput, etc.), where only network experts are able to settle them correctly.

The advantage of model-driven approaches are that the models do not require empirical data and the behavior of a system can be derived solely (at least in theory) by a set of ‘hard’ parameters, like the bit rate of a channel, input packet rates of the applications, connectivity information (topology and conflict graph), and so on.

Model-driven argumentation itself needs confirmation by real-world (i.e. empirical) validation, or otherwise, it does not produce any insight. In this thesis, therefore, we pursue a model-driven approach to the modeling of latency in wireless multi-hop networks, while our models have to be empirically verified by real-world measurements. This way, we seek to deliver modeling techniques for the prediction of network behavior that have a methodological basis and do not require empirical expertise for their adoption. This, however, is a goal that we can not fully reach within this thesis alone, and the validation necessarily must be understood as a partial approach, since it must prove to hold in recreated scenarios, too.

For the analytical modeling, we adopt two common modeling techniques that are commonly used in computer systems performance analysis, which are Markov models and queueing theory. Both are statistical methods that base on probability theory. The resulting approach, thus, is as well of statistical nature.

3.2 Evaluation Methods Used

Scientific theories have to be evaluated to prove their feasibility in the real world. Three methods can be used here [63]: Simulation, testbed experimentation, and case studies. In this thesis, we use all three for specific parts of our study.

3.2.1 Discrete-Event Simulation

Simulation uses a model of the studied system that can be described both in terms of its internal state and of how the state changes over time. If the time advances continuously, as for differential equations, for example, then one speaks of continuous simulation. Otherwise, it is discrete-event simulation. Discrete-event simulation is the primary tool to simulate computer communication systems [64].

In discrete-event simulation, the *state* of a system is a well-described set of properties with exact values at any given time t. The state can only change when an *event* occurs. An event is associated with an action (i.e., a state change), which takes effect at the occurrence time. Events can be generated either during another event execution, or based on time, e.g. generated periodically or at a given time instant. Upon generation, events are placed in a queue called *future event list*, where they are kept until they take effect. The occurrence time of an event must always be in the future at the time an event is generated.

Since the system state does not change in between two events, it is sufficient for the 'execution' of the simulation to iteratively take the next event from the future event list and execute the associated actions with that event. The simulation time is automatically advanced to the time point in which the next event takes effect. A simulation must specify all state variables, event generation rules, and actions, and place the first event in the future event list to start a simulation. The simulation may either end after a certain elapsed time or after a designated termination event. The consecutive list of iterated states (the simulation history) is the simulation data generated, which can then be evaluated.

When simulating random processes, such as queueing behavior or random noise, discrete-event simulations always uses pseudo-random number generators for execution. A well-defined simulation (with all actions, generation rules and state definitions completely specified), initialized with the same random seed on every execution, will always repeatably produce the same output every time it is executed. This way, simulations can ensure repeatability of experiments, which is a necessary property for scientific evaluation. All simulation tools allow for serialization of experiments, system model description, random number generation, and experimentation results, so that results and experiments can be shared easily between research groups.

For the discrete-event simulations in this thesis, we use the tool OMNet++ [65].

3.2.2 Testbed Experimentation

In contrast to simulations, testbeds do not rely on models of systems, but take a real-world implementation of the system (or, at least, parts of it) to generate test data. The advantage is that the experiments always produce real-world behavior, whereas simulations necessarily suffer from an unknown bias from the model uncertainty. A drawback from real-world experiments, however, is that they are mostly not repeatable. For network simulation, for example, even the hardware or drivers can affect the experiment outcome. Similarly, for wireless transmissions, the surrounding of the experiment can affect its outcome. It is often not possible to create an experiment without any interference from the outside within the lab, and if so, the scientific value of such a confined experiment under lab conditions is often debatable as well. To solve this problem, testbeds create a well-defined experiment setup, where hardware and software is standardized, such that consecutive experiments will produce similar outputs. A testbed, therefore, solves the problem of repeatability for gathering real-world experiment data [63].

Within this thesis, we use the MIoT-Lab [66] as a testbed for our network experiments.

3.2.3 Case Studies

Case studies, as well as demos, are known to be not repeatable at all, and therefore, their scientific value is often limited. However, in terms of showing the applicability of certain methods, they help to clarify some implementation-related questions that often arise with new scientific ideas. We use a case study in Chapter 10 to support the assumptions that we make throughout the formalization and modeling parts of this thesis, as well as to demonstrate the use of our Haptic Communication Testbed at the Otto-von-Guericke University of Magdeburg (OVGU-HC).

3.3 Code of Experimentation

For this thesis, we rely both on testbed experiments and on simulations to prove our hypothesis. We do not conduct any analytical validation of our software framework, or of our implementations of the Tactile Control Function. Our hypotheses are principally either of two kinds: either we want to show that our analytical models (in Chapter 7 and 8) match the behavior of a real-world network, or we show that an implementation of some software concept fulfills some service qualities (in Chapter 9 and 10). Since both testbed and simulation experiments provide the opportunity for easy repetition, we are always able to collect enough samples to reduce the error margins so that the benefits of our solutions become clear against some baseline reference.

3.4 The MIoT-Lab

The Magdeburg Internet of Things Lab (MIoT-Lab) [66, 67] is a testbed for application software and communication protocols for the Internet of Things. It consists of a set of 200 homogeneous nodes which form a multi-hop network on different frequencies and with different network technologies. The nodes each consist of an x86-board with a Linux operating system, and an IoT-board running RIOT OS [68, 69]. The multi-hop part is formed by IEEE 802.11n adaptors that operate from the x86-board, while the wireless sensor network (WSN) part can be configured from IEEE 802.15.4, Bluetooth Low Energy, LoRA, or a sub 1-GHz transceiver.

The nodes are centrally managed by a Testbed Management System (TBMS) that offers a web interface for experiment specification, deployment, scheduling, and execution. Experiments can be configured with a domain-specific language called DES-Cript [70], which allows topology definition, firmware specification, experiment design, and storage of experiment results. Figure 3.1 shows an overview of the MIoT-Lab installation within the building 29 of the Otto-von-Guericke University Magdeburg. Figure 3.2 shows the node hardware.

As part of the MIoT-Lab, we introduce the OVGU-HC in this thesis. OVGU-HC consists of a tactile robotics lab and a teleoperation workspace which can be used for evaluation of Tactile Internet Applications. It supports a wide range of hardware setups, from teleoperation

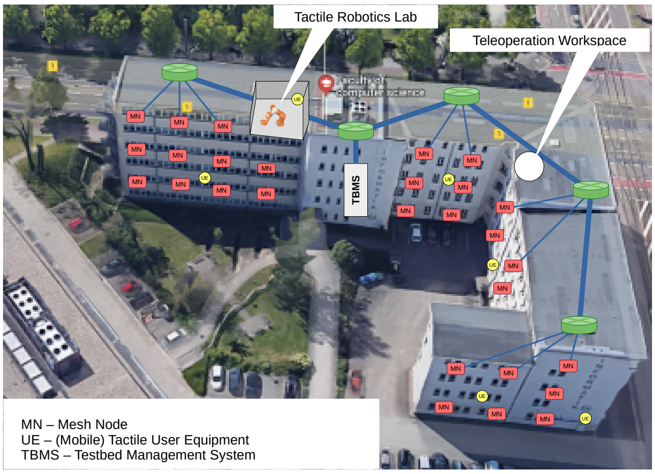

Tactile Robotics Lab
Teleoperation Workspace
TBMS
MN
UE
MN – Mesh Node
UE – (Mobile) Tactile User Equipment
TBMS – Testbed Management System

Figure 3.1: Location of the MIoT-Lab nodes.

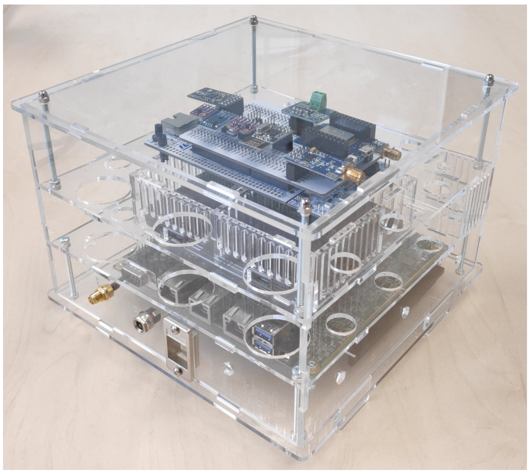


Figure 3.2: A MIoT-Lab node with its internal cabling removed.

of industrial robots to the deployment of NCSs over different networks such as the Internet. We introduce the OVGU-HC in Chapter 10.

3.5 OMNet++

OMNet++ is a simulation tool for discrete-event simulations of various kinds, developed in C++. Its focus is on simulations of computer networks, queueing networks, communication protocols, and computer systems, and is widely used in academia, research, and industry. It provides modularity for experiments, being able to encapsulate functionality in modules for re-use. Experiments can be assembled from a large pre-defined module library, or by programming new modules in C++, while the experiment definition is done in a custom text-based file format called NED. Experiments are compiled with a C++ compiler into an executable file format, which can be chosen to support a wide range of computer systems, including workstations and servers. For the experiment definition as well as module development in C++, an integrated development environment is provided, which is derived from the Eclipse platform.

We use OMNet++ in conjunction with the INET Framework [71], which is a collection of OMNet++ modules built specifically for in-depth simulation of IP-based network technologies. It offers simulation of the full functionality of all network layers, including transmission models, and the implementation of network protocols. It supports Ethernet, IEEE 802.11 and IEEE 802.15.4, as well as generic wireless and wired protocols. Implementations for protocols like TCP, UDP, RTP, IPv4, IPv6, PPP, OSP, and BGP exist and can be extended. We make excessive use of the `Ieee80211Interface` for simulating WiFi networks within this thesis.

Simu5G [72] is another collection of modules for OMNet++ targeted for 5G New Radio simulation. Other libraries further extend OMNet++ with regards to V2X communication, real-time Ethernet, or peer-to-peer networks.

3.6 Graph-based Network Modeling

The main method of analysis in this thesis is graph-based modeling of computer networks. Network hosts and routers are represented as nodes of a graph, and communication links are represented as edges between nodes. Such a model often both represents the topology of the Network Layer and the behavior of transmission protocols on the Data Link Layer. The clear distinction between the two layers in terms of the computer networking theory is often lost. This is the case in our analysis, too, as we model the behavior of the communication links based on the respective Layer-2 protocols, but also model the queueing behavior of the nodes in terms of network congestion. However, the distinction between the layers remains intact throughout all steps of analysis. It is thus not a clear cross-layer approach, but an informed analysis that uses a grey-box approach. This means, that protocols can still be exchanged freely and without conflicts in interfaces.

A graph-based model consists of a set of *nodes* $\mathcal{V}$, with $|\mathcal{V}| = n_{\mathcal{V}}$, and a set of *edges* (links) $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, which is a set of node pairs. An edge expresses a relation between two nodes, which can either be a neighborhood relation, such as node v and node u can communicate over a given Layer-2 communication protocol. Edges are not necessarily symmetrical, which means that $(v, u) \in \mathcal{E}$ does not necessarily imply $(u, v) \in \mathcal{E}$. Properties of nodes and edges can be expressed by functions $f : \mathcal{V} \to \mathcal{A}$ or $g : \mathcal{E} \to \mathcal{B}$, where f and g are properties of nodes or edges, $\mathcal{A}$ and $\mathcal{B}$ are sets of property values, respectively. For example, $\mathrm{load}(v) : \mathcal{V} \to \mathbb{R}$ represents the network load at node v.

We call a graph $(\mathcal{V}, \mathcal{E})$ a *network topology*, if the relation $\mathcal{E}$ represents the Layer-2 neighborhood of the nodes $\mathcal{V}$ in a given network. In addition, we denote the set of neighbors of a node v, that are adjacent to node v in the network topology, as $\mathcal{C}_v$. For wireless communication, $\mathcal{C}_v$ is the set of nodes that are within the communication range of v.

Another relation is defined by the detection range of the wireless transceivers. The detection range is the set of nodes around a given node v to which a direct communication is not possible, but v is able to detect their active sending. It is also called *carrier sensing range*, since v can detect the radio carrier frequency but it may not be able to decode a transmitted signal. We call the set of nodes around v within carrier sensing range as $\mathcal{D}_v$, and the set of edges that define the detection range relation as $\mathcal{E}_{\text{con}}$. The graph $(\mathcal{V}, \mathcal{E}_{\text{con}})$ is called the *conflict graph* of that network. Often, the relation $\mathcal{E} \subseteq \mathcal{E}_{\text{con}}$ holds, so the conflict graph is an extension of the network topology, but we do not strictly require this axiom.

Figure 3.3 illustrates the relations between the network topology, the conflict graph, and the detection and communication ranges $\mathcal{D}_v$ and $\mathcal{C}_v$.

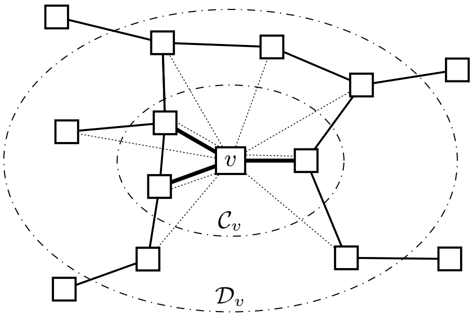


Figure 3.3: Network graph highlighting the carrier-sense range $\mathcal{D}_v$ and connectivity range $\mathcal{C}_v$ of a node v. Edges incident to v in the conflict graph of the network are shown as dotted lines.

A *path* through a network is defined as a set $\mathcal{P} \subseteq \mathcal{V} \times \mathcal{V}$ of tuples, $\mathcal{P} = \{(v_0, v_1), (v_1, v_2), (v_2, v_3), ..., (v_{l-1}, v_l)\}$, $|\mathcal{P}| = l$, where each $(v, u) \in \mathcal{P}$ must be part of the network topology, $(v, u) \in \mathcal{E}$. l is called the *hop count* of path $\mathcal{P}$. The nodes v_0 and v_l are the *source* and *destination* hosts with a full TCP/IP stack up to the Application Layer.

3.7 Formal Notation and List of Symbols

To avoid confusion between different notations, and to ease reading, we use the following rules for mathematical notations throughout this thesis.

- Scalars are set in lower case letters, e.g. $x, \lambda, \hat{a}$.
- Random variables are set in capital letters, e.g. A, B, Q.
- Sets are set in capital calligraphic letters, e.g. $\mathcal{C}, \mathcal{V}, \mathcal{E}$.
- Matrices and vectors are set in bold capital or lower case letters, respectively, e.g. $\mathbf{M}, \mathbf{x}$.

Symbol	Description
General notation:	
$\Pr\{X\}$	Probability of event X
$f_X(x)$	Probability density function (PDF) for variable X, $f_X(x) = \Pr\{X = x\}$
$F_X(x)$	Cumulative distribution function (CDF) for random variable X, $F_X(x) = \Pr\{X < x\}$
$\mathrm{E}[X]$	Expected value for random variable X
$\mathrm{Var}(X)$	Variance of random variable X
$\mathrm{Cov}(X, Y)$	Covariance between random variables X and Y
X_a	a-quantile of random variable X
$\delta(x, y)$	Kronecker delta: $\delta(x, y) = 1$ if $x = y$, or 0 otherwise
ϵ	One-vector with all elements equal to 1
Network and traffic models:	
$\mathcal{V}$	Set of nodes
$n_{\mathcal{V}}$	Number of nodes, $n_{\mathcal{V}} = \lvert\mathcal{V}\rvert$
$\mathcal{E}$	Set of edges of the topology graph
$\mathcal{E}_{\mathrm{con}}$	Set of edges of the conflict graph
$\mathcal{D}_v$	Detection range of node v, $\mathcal{D}_v \subseteq \mathcal{V}$
$\mathcal{C}_v$	Communication range of node v, $\mathcal{C}_v \subseteq \mathcal{V}$
t_{slot}	Slot time
s	Payload size
h	Header size
r	Packet rate
Application model:	
t_{SSS}	Steady state share
t_{trans}	Transition time
t_{pcv}	Perceived delay
t_{e2e}	End-to-end delay
e	Error metric
n_{f}	Number of flows
l_i	Number of hops in flow i
Random variables (r.v.) of the probabilistic model:	
$T_{\mathcal{P}}$	R.v. of delay on path $\mathcal{P}$
$T_{u;v}$	R.v. of single hop delay between nodes u, v
Q_v	R.v. of queueing delay at node v
B_v	R.v. of contention delay at node v
A_v	R.v. of arbitration delay at node v
$M_{u;v}$	R.v. of the number of transmission attempts from node u to v
$D_{u;v}$	R.v. of busy time
S_v	R.v. of service time at node v
N_v	R.v. of queue length at node v
Divide-and-conquer throughput model:	
$\mathrm{On}, \mathrm{Off}$	Node regimes indicating wether a node has a waiting packet in its tx queue
$b_i(v)$	Regime of node v in subnetwork i, $b_i(v) \in \{\mathrm{On}, \mathrm{Off}\}$
$\mathbf{b}_i$	Subnetwork i, $\mathbf{b}_i = (b_i(v_1), ..., b_i(v_{n_{\mathcal{V}}}))$, a regime attribution for all nodes in a network of nodes $\mathcal{V} = \{v_1, ..., v_{n_{\mathcal{V}}}\}$
$\mathcal{B}$	Set of subnetworks, $\mathcal{B} = \{\mathbf{b}_i\}$, containing all valid regime combinations
β_i	Occurrence probability of subnetwork i

s_k	Sending state k, $s_k \in \{0,1\}^{n_\mathcal{V}}$, indicating which nodes are currently actively sending
$\mathcal{S}$	Set of sending states $\{0,1\}^{n_\mathcal{V}}$ for the set of nodes $\mathcal{V} = \{v_1, ..., v_{n_\mathcal{V}}\}$
$s_k(v)$	Activity of node v in sending state k, $s_k(v) \in \{0,1\}$
$\mathcal{S}_i$	Set of valid sending states of subnetwork $\mathbf{b}_i$, $\mathcal{S}_i \subseteq \mathcal{S}$

ME queueing model:

$\mathbf{S}^{(v)}$	Subsystem of node v, $\mathbf{S}^{(v)} = \left(\mathcal{Q}, \mathbf{p}^{(v)}, \mathbf{P}^{(v)}, \mathbf{q}^{(v)}\right)$
$\mathcal{Q}$	Set of phases, $\mathcal{Q} = \mathcal{B} \times \mathcal{S}$
$\mathcal{Q}'$	Set of valid phases, $\mathcal{Q}' \subseteq \mathcal{Q}$
o	Number of phases, $o = \|\mathcal{Q}\|$
$p_k^{(v)}$	Initial probability of phase k for subsystem $\mathbf{S}^{(v)}$, $p_k^{(v)} \in [0,1]$
$\mathbf{p}^{(v)}$	Vector of initial probabilities, $\mathbf{p}^{(v)} \in [0,1]^{1 \times o}$
$q_k^{(v)}$	Exit probability of phase k for subsystem $\mathbf{S}^{(v)}$, $q_k^{(v)} \in [0,1]$
$\mathbf{q}^{(v)}$	Vector of exit probabilities, $\mathbf{q}^{(v)} \in [0,1]^{1 \times o}$
$p_{k;l}^{(v)}$	Transition prob. from phase k to phase l for subsystem $\mathbf{S}^{(v)}$, $p_{k;l}^{(v)} \in [0,1]$
$\mathbf{P}^{(v)}$	Matrix of transition probabilities, $\mathbf{P}^{(v)} \in [0,1]^{o \times o}$
$\pi_k^{(v)}(t)$	Probability of phase k after t steps

Signal propagation model:

capacity(v)	Channel capacity at node v
load(v)	Channel load at node v
L_{SNIR}	Signal-to-noise ratio
c_M	Speed of light in medium M
$d(u,v)$	Euclidean distance between u and v

PART II

An Open Tactile Internet Architecture

CHAPTER 4

Review of the Tactile Internet Vision

In this chapter, we review the current technologies, standards, and literature about the Tactile Internet. The Tactile Internet has a relatively short history that effectively started in 2016 with a seminal paper by Simsek et al. [4] from the 5G community. Some technologies, however, have been developed for some decades already and can be reflected to gain insights into future developments. Our goal is to develop an open architecture for the Tactile Internet that is compatible with the current structure of the Internet. Approaches have been discussed already in the literature, and also an IEEE workgroup has been formed to work towards standardization. We give an overview of all these approaches and discuss them at the end of this chapter.

4.1 The General Application Scenario

We have already motivated the Tactile Internet with a telesurgery application in the first chapter that requires 1 ms round-trip latency. According to the classic perspective taken by application engineers, ideally, the network would simply offer low-latency communication as a non-functional aspect in terms of a service quality agreement. Given such an agreement, the network conditions would allow such an application to operate. This perspective appears as a provider-consumer model, where the network is a service provider fulfilling some given metrics in order to allow a consumer (a Haptic Application) to function properly. The Haptic Application simply *consumes* network resources. According to this model, new resources can be added by providing some kind of expense, either better networking hardware or new communication channels.

This classic perspective, however, is now no longer feasible since Haptic Communication hits the physical limits of network communication. A network can not just provide 1 millisecond of latency to any point in the world in form of a service agreement. As a result, the application must adapt to the implications of the network and must alter its communication requirements. For the telesurgery example, the solution was the installation of TSEs in a place where the physical limitations allow for low-latency communication. This is a step where the application has to adapt to its own communication needs, specifically by introducing more complexity into its communication structure. In other words, the application has to adapt to the network, instead of the other way round.

This development becomes necessary since communication latency is a concern. In the past, mostly bandwidth and reliability requirements were a concern, which could be solved by investing more and more resources on the network side. The producer-consumer view was sufficient for these requirements. Now the network topology must come into the focus of application developers, who now have to be concerned about their server placement and the more complicated application structure that is connected with this development. This *1 ms-problem* is thus a concern of the proper distribution of platform infrastructure across the Internet, which requires a proper understanding of the underlying network structures and their influence on the QoE.

From this perspective, applications with endpoints that are far away from each other must have a different structure than applications that only communicate locally. We have a separate look into these two circumstances in the case of Haptic Communication.

4.1.1 Haptic Communication in Local Area Networks

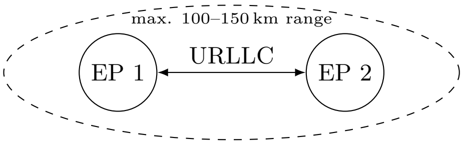


Figure 4.1: Haptic Communication between endpoints in a Local Area Network.

When considering Local Area Networks, as in Figure 4.1, the communication distances between endpoints are small and the propagation delays are usually far below the 1 ms mark. In this situation, the classic QoS-oriented approach can be used, and Haptic Communication can be handled via URLLC. The network can offer a certain latency as a service agreement to applications, and it is also physically able to fulfill this agreement. The locality occurs only in lower-tier networks, such as Access Networks, that have limited spatial extent and thus the distances are small. As a benefit, Access Networks are often managed by a single authority, which makes it easier to employ the necessary capabilities for QoS-aware communication services. The 1 ms-problem can be solved by communication protocols, for example by resource reservation or with traffic shaping (prioritization of low-latency traffic).

We already have calculated that the 1 ms latency bound for Haptic Communication related to a maximum distance of about 100–150 km between peers due to the limiting propagation delay (see Section 2.3.1). This is also the typical scenario that has been addressed with the IMT-2020 specification for 5G, which requires 1 ms round-trip latency in the 5G user plane. Technical solutions for this scenario exist, and as we demonstrate later in this thesis, also WiFi networks can offer latencies within this range. For Haptic Applications, it is favorable when the underlying network can just provide the required QoS constraints, since no adaptations to the communication model are needed. All communication models, such as peer-to-peer, or client/server, are possible, and no extra network entities, such as the TSEs in our initial telesurgery scenario, are required.

4.1.2 Haptic Communication on the Intercontinental Scale

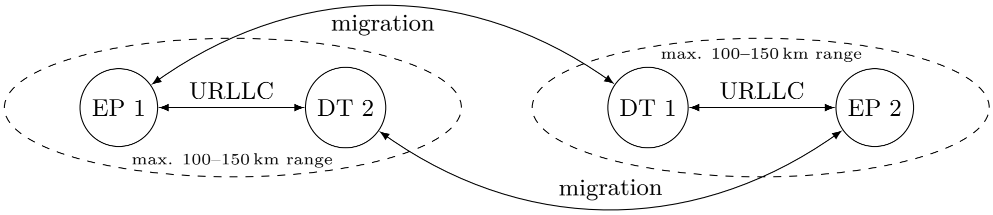


Figure 4.2: Intercontinental Haptic Communication with Digital Twins.

On the intercontinental scale, the perspective must change, as shown in Figure 4.2. The physical limits do not allow for network guarantees between the endpoints. An application must be able to *migrate* towards the respective peer to reduce propagation delay, which adds a high amount of complexity that is also application-specific. Software frameworks can provide assistance, but general approaches that are transparent to the application are unlikely to succeed. Caching, for example, which is a common strategy to reduce response time on the

Internet, can not be used for Haptic Communication, as it is only applicable to static content. Artificial Intelligence may be the only method to provide the necessary prediction horizon but must be designed specifically for each application.

However, many common functions may be similar across Haptic Applications. Haptic Communication is a known subject and has specific requirements. Haptic Coding is a task that must be performed for almost every application. Also, the isochronous shape of Haptic Communication allows for a specific design of applications. As recent proposals for system models in literature show [6, 38], the communication endpoint is almost ever a control loop, which means that the control loop as such can be regarded as an application primitive for Haptic Applications in general. Other primitives can be haptic display technologies and tactile information databases [6]. Specific frameworks can offer such mechanisms to application developers in a PaaS manner, where the application logic can be built from a set of primitives and with certain infrastructure components such as storage, configuration, semantics, and migration support.

A similar case where the locations of hosts affect the performance of a system is that of CDNs. CDNs, too, migrate content closer to clients in order to reduce the response time and optimize network load. However, these services only duplicate static media content. As such, they behave relatively transparent, even to application developers, and can easily be distributed as needed without affecting the basic client/server communication principle of the application in question. With Haptic Communication, this is not the case anymore, as the communication is dynamic and also bi-directional. The infrastructure thus is not transparent and applications must be specifically designed to have more than just two endpoints.

4.1.3 Conclusions

In summary, we can conclude that the legacy view of modeling communication networks as mere service providers must be abandoned for the Tactile Internet. The necessary new perspective must be that the network influences the application, and the application must adapt to network conditions. Applications now have to consider the network as a subsystem of equal importance as the application roles. Therefore, we introduce *network roles* as a part of each application, which interact with the classic application roles (client/server, master/follower, peer-to-peer). The network roles must, at least, represent the two argued ones ('LAN' and 'intercontinental network'), as well as the influence of the network on the applications. We investigate these network roles further in Chapter 10 when we develop an application framework for Tactile Internet Applications.

4.2 Relevant Enabling Technologies and Research Fields

The Tactile Internet has been driven by recent technological advancements, especially by the fifth revolution of telecommunication networks (5G). The development towards a standardization effort has been based on the original idea from the group of Fettweiß et al. [4], where they introduce the Tactile Internet as an evolution of cellular mobile telecommunication systems. But also other developments have started to focus on low-latency and highly reliable communication systems.

Although 5G and its continuous evolution (currently 6G) are often stated to be the key driving technology behind the Tactile Internet [4, 24, 27, 42], we find that mobile telecommunication networks will cover only the local solution.

All recent ideas on haptic teleoperation, networked control, or high reliability that go along with the Tactile Internet vision are based on numerous technologies that are currently pushing from the state of a scientific vision towards reality. The rolling evolution of cellular telecommunication networks is only one of them. We do not further cover 5G or 6G as technological foundations, but try to take a broader view, without assuming a single network technology or infrastructure as a prerequisite. We consider the following developments as essential for the development of the Tactile Internet: Digital Twins, IP-based networking, Next Generation Access Networks, and the development of a Haptic Physical Layer.

4.2.1 Digital Twins

A Digital Twin is a digital representation of a real-world object, or process, that exists in a state of constant data exchange with its real counterpart. The concept was first described in the context of Product Lifecycle Management [73], but can be extended to virtually any real-world entity, including buildings, technical or non-technical objects, and persons. The concept, as visualized in Figure 4.3, is based around an object that exists both in a virtual space and in the real space as a real object. Both virtual and real objects represent the same entity and thus have to be synchronized permanently, as both the digital representation and the real object can change.

The rationale behind a Digital Twin is the availability of a full description of an object for a specified purpose in order to eliminate the necessity to be in close proximity to that object [7]. It steps into the role of a physical object whenever the real object is not available, too distant, or otherwise restricted from physical access. A digital object is both migratable and multipliable arbitrarily and imposes no restrictions in terms of its use. The Digital Twin can act as a replacement of a real object for a given purpose, when the real object is either not available, or even not yet existing.

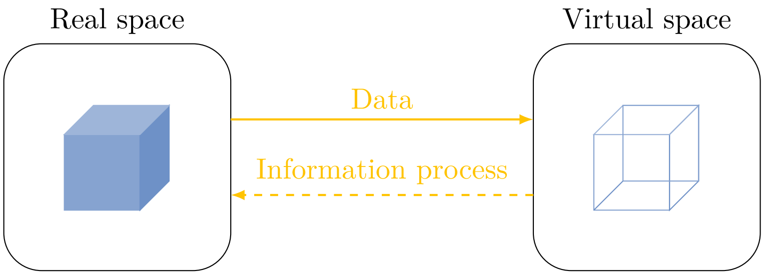


Figure 4.3: Concept of a Digital Twin [73].

Two directions of interactions between the two spaces are possible, as shown in Figure 4.3. The data relation manifests when the virtual object is created or updated as an initial ‘digitalization’, or to represent changes to the real object. This step can happen in two different stages. For a *Digital Twin Product*, the virtual object is created before the real object to plan its production process. A Digital Twin Product can consist of a set of requirements, fully annotated 3D model, bill of materials, bill of processes, bill of services, and bill of disposal [7]. For a *Digital Twin Instance*, in contrast, the real object always coexists with the virtual object. It can consist of all aforementioned information of a Digital Twin Product, and, in addition, can contain a history of service interactions that have been invoked on the instance.

Digital Twins can be used either for predictions of a real object, or for interrogation of its current state of its history. Here, the living interaction between virtual and real space is vital for its purpose, and the synchronization between real and virtual space requires new approaches for the current understanding in data processing. Full Digital Twins, where both directions of interactions are present and fully developed, do not exist yet. One precursor to the Digital Twin is the *Digital Shadow*, that mirrors the state of a physical object, but does not implement the backward information link. Another precursor is the *Digital Model*, that acts as a production template, but does not include a feedback data link from the physical object to the virtual model.

As the physical distance between endpoints is one of the key concerns in the Tactile Internet, the Digital Twin concept can serve as a solution to the 1 ms-problem, which is more or less a proximity problem. However, for the purpose of bridging distances between endpoints, the synchronization problem for Digital Twins has to be solved. Concepts in this direction have been proposed already. For example, with Model Mediated Teleoperation [32], the teleoperator and its environment are digitalized by means of a 3D model. It is enhanced with surface data to simulate interactions between the teleoperator and the environment, such that the occurring moments and forces can be re-created by the operator interface. Proposals where Model Mediated Teleoperation is used as a general solution for the Tactile Internet have already been made [74].

4.2.2 Ongoing Service Unification and Transition Towards IP Based Networking

The Internet went through many evolutionary steps to its current shape, while adopting, refining, and consolidating different technologies over time. In the mid 2000s, under the term New Generation Networks, a transition towards an all IP-based infrastructure began, and many legacy systems derived from telephone and cable networks were replaced by packet-switched networks. One of the major concerns of 5G was to unify both the Core Network and the Radio Access Network and transform the remaining telephony services to Voice over IP (VoIP). However, in 5G, still three different hardware modes (eMBB, mMTC URLLC) coexist [11, 75] that serve different applications, and that can not be unified in near future due to their conflicts of interest. With 6G, new modes likely come in addition, for example, satellite communication, and the integration of WiFi and Bluetooth networks.

The integration of different hardware modes, on one hand, enables further integration and unification of network services, but it also introduces integration problems due to different tradeoffs made for the individual modes. A seamless integration, allowing for seamless handoff and an end-to-end QoS mapping, is hardly possible if different infrastructures exist for different services, different use-cases, and within the networks of different service providers.

As a possible solution, customers can rent capacities from ISPs for horizontal integration through network slicing. Network slicing allows for capacity planning in terms of latency and throughput, thus increasing network cohesion for special purposes. This development opens up new possibilities for customers, as data centers and endpoints can grow together, such that, finally, whole network infrastructures transform into computing networks, rather than being sole transmission networks [75].

With 5G, network slicing is only available within the Core Network, but not for Access Networks, leaving a gap for handoff situations and border traffic. We can conclude, that the state of the Next Generation Networking vision is not yet reached and that the ongoing developments will rather have a character of a rolling integration over the next few decades.

4.2.3 Next Generation Access Networks

Access Networks currently undergo substantial developments, as the biggest part of system unification, and also the development of new Physical Layer protocols, continuously affect the Edge Networks. The permanent transformations from a circuit-switched network (1st Generation) to the recent 5G-NR standards demonstrate its fast-pace development.

Two developments in near future are important to Access Networks in particular: *densification* and the adoption of *multi-hop technologies*. Densification refers to an increase in the number of connected devices within a confined area. As for the specification of IMT-2020 [11], the required device density for 5G is specified with one million devices per square kilometer. The increased device density imposes big challenges for infrastructure and, finally, must lead to a reduction in cell size. A typical microcell, for example, has a range of fewer than two kilometers, while a femtocell can have a range of just around 10–20 m. For urban applications, or applications within campuses, industrial complexes, or office buildings, such cells become more and more important. This is supported by the fact that the required bandwidth per device has increased over the last mobile generations, which leads to the inclusion of higher frequency bands that have shorter ranges.

Densification leads to a higher coverage for network infrastructure. Typically, a Radio Access Network is divided into two parts, the *fronthaul* and the *backhaul* part. The fronthaul system is responsible to connect the end-user devices (*user equipment*) to the network infrastructure. In WiFi, the fronthaul network consists of the Access Points and their Basic Service Sets (BSSs), while for cellular networks, the fronthaul network is covered by the Base Stations. The backhaul network, on the other hand, connects these stations to the next gateway that has access to the Core Network. With shrinking cell range and increasing cell size, more research focus has to be put on the backhaul networks, as they continuously grow in size and numbers.

Multi-hop topologies have been identified as a necessary development to control the growing complexity infrastructure of the network infrastructure. This development is known under the term Self-Organizing Network (SON) [76], which covers means for the reduction of service cost for Radio Access Networks. Multi-hop networks are available mostly with ISM-band technologies only (WiFi, Bluetooth, IEEE 802.15.4, for example), but the ongoing development of 5G-NR also envisions multi-hop technologies for 5G, and can be expected to play a core role in 6G. Current backhaul technologies rely mostly on point-to-point line-of-sight connections using proprietary protocols [77]. First standardization for relay links was made in LTE Release 10, however, more standardization is expected for 5G-NR, where mm-wave frequencies are available to serve in the backhaul as well as in the fronthaul. A respective work item called Integrated Access and Backhaul (IAB) has been included in Release 16 and specification is expected to start with Release 17.

4.2.4 A Haptic “Physical Layer”

The Tactile Internet requires the development of new haptic input and display devices. Haptic Communication can not be expressed in terms of classical human-machine interfaces, like the computer mouse and keyboard, monitors, microphones, and loudspeakers. First sensations from the haptic modality could be displayed with early force-feedback controllers (joysticks). Modern haptic input devices can be compared to joysticks in the sense that they present sensations by projection to an input device, but they have been extended to precisely display forces and torques in more than one dimension.

For the tactile modality, surfaces and tactile displays can be used to simulate the surface structure of an object [6]. They can also display temperature information. As an input device, tactile skin [78] can serve as a replacement for the sense of touch.

Haptic Communication can accelerate the development of deep immersion virtual worlds. The Tactile Internet can therefore also profit from developments of new types of Virtual Reality (VR) and Augmented Reality (AR) technology, as well as Head Mounted Displays. For a true “Haptic Physical Layer”, these technologies need to be integrated as new display and input devices into the World Wide Web, similar to the recent integration of Smartphones, Wearables, and eBook-Readers.

4.3 The IEEE 1918.1 “Tactile Internet” Draft Standard

The IEEE has established a Working Group for the Tactile Internet, the IEEE 1918.1 Standards Working Group. Its purpose is the surveying of requirements, collection of technologies and concepts, and standardization of protocols for the Tactile Internet. An early summary of their work has been published in a 2019 paper [5]. In general, the IEEE 1918.1 focuses more on the Application Layer and on network infrastructure questions. Layer 2 and 3 issues are not explicitly considered.

There will be a “baseline standard” IEEE 1918.1 for the introduction of notation, architectural considerations, and use-cases. A sub-standard, currently work-in-progress, is the standard IEEE 1918.1.1 “Haptic Codecs for the Tactile Internet”. While the standardized protocols are intended to work “on top” of existing Internet standards and technologies, 5G mobile telecommunication networks are seen as a key enabling technology for the Tactile Internet [5]. The IEEE 1918.1 standards affect the Tactile Internet infrastructure, Tactile

Internet Applications, and, as envisioned, also Application Layer protocols. The main concepts of the standard comply with the IMT-2020 specification for 5G telecommunication technology.

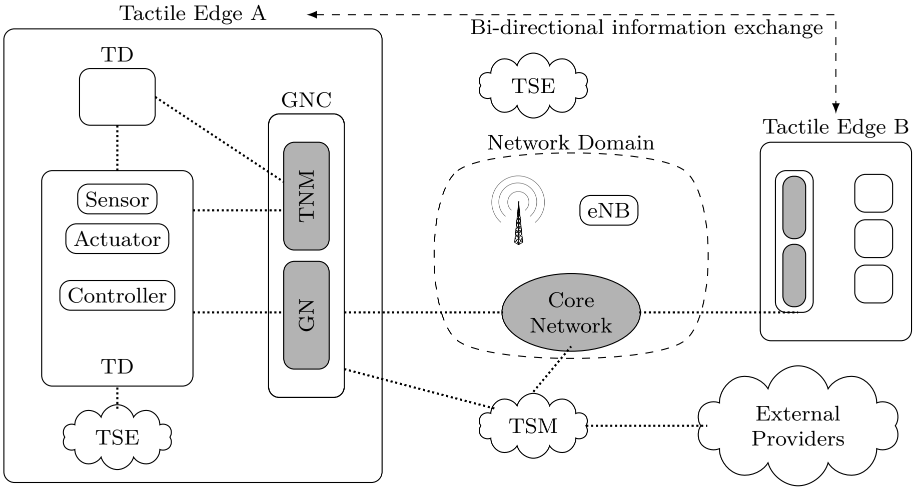


Figure 4.4: Tactile Internet Reference Infrastructure, adapted from [5].

The core infrastructure of IEEE 1918.1 is focused around Access Networks and is designed after cellular telecommunication standards. It is depicted in Figure 4.4. A Tactile Edge consists of a set of Tactile Devices (TD) which consist of sensors and actuators. They are interconnected with a Gateway Node Controller (GNC), which acts as the network interface towards the Access Network. The GNC itself consists of a Gateway Node (GN) and a Tactile Network Manager (TNM), which separates both the networking and the management part (QoS control, etc.) of the network interface. The TSEs can be either part of the Tactile Edges themselves or part of the Access Network.

The actual network is not part of the specification, which role likely falls to 5G. It is subject to Next Generation Networks to act as the transport medium to fulfill the requirements of applications. The IEEE 1918.1 standard is, however, explicitly agnostic to the actual networking solution, and all future IEEE 1918.1 protocols will make no assumptions on the network stack that is used [5].

As the standardization is still in an early phase, it is not clear how the approach to manage the data traffic for the Tactile Internet will look like. It is also not clear if the standard will be fully or partly adopted by the Internet community. But at this stage, it already shows that many lessons learned from former mobile telecommunication generations have been integrated. Especially the openness of the network domain is a key factor for driving innovation within the community. However, the tight bonding to mobile telecommunication networks might come as a hindrance to IEEE 1918.1. The liberally operated Internet might not obey the strict interface definitions proposed so far by the standard, and chances are high that alternative approaches emerge. The development could be similar to the liberally developed TCP/IP reference architecture, which displaced the more strictly standardized ISO/OSI model.

4.4 Relevant Development of Open Protocols

The IEEE 1918.1 standards have been inspired by a set of more specific communication standards, which we want to introduce here shortly. We show that the most developed work on wireless technologies has been conducted on 5G mobile telecommunication standards. For 6G, these standards can be regarded as integrated. Wireless multi-hop networks based on WiFi (IEEE 802.11), however, are yet underrepresented in literature.

The Internet has always relied on standards for its growth, ensuring its interoperability. Without standardization, the Internet probably would not exist in its present global state.

The same development will continue in the future, so it is no surprise that many new concepts, like the Tactile Internet, are standardized as soon as their idea materializes enough to foresee requirements and technical possibilities. In order to survey the standardization work, we first present the ongoing efforts of standardization regarding the Tactile Internet and Haptic Communication. After that, we identify missing gaps in the constellation.

4.4.1 IEEE Time-Sensitive Networks

Along with the mobile telecommunication technologies moving towards low-latency and ultra-reliable communication, also the Institute of Electrical and Electronics Engineers (IEEE) and Internet Engineering Task Force (IETF) conduct standardization efforts in that direction.

The IEEE has a specialized Task Group within their 802.1 Work Group working on internetworking standards regarding DetNets and URLLC. The concept is called Time-Sensitive Networks (TSNs). A recent survey was published in 2019 [79]. Since the 802 section of the IEEE covers all other networking protocols like Ethernet, WiMax, and WiFi, the proposed standards can be regarded as an amendment to those technologies. The TSN Task Group has released several accompanying standards, like 802.1Q (TSN for Bridges and Bridged Networks), 802.1AB (Link Layer Connectivity Discovery), 802.1AS (Timing and Synchronization), 802.1AX (Link Aggregation), 802.1BA (Audio Video Bridging Systems), 802.1CB (Frame Replication and Elimination for Reliability), and 802.1CM (TSN for Fronthaul). Many smaller sub-standards have been published already under these categories, extending their coverage to specialized networks and application scenarios, like, for example, metropolitan area networks. TSN works with the concept of *flows*, and so the standardization is divided into five related categories: the overall flow concept, flow synchronization, flow management, flow control, and flow integrity.

The TSN flow concept is characterized by QoS properties like the required throughput, required latency, and similar. These properties are always bound to a specific traffic class, which is identified by the 802.1Q Virtual Local Area Network (VLAN) ID of each frame.

The scope of the IEEE 802.1 standards is restricted to bridged networks and thus to the Medium Access Control (MAC) level. Traffic classes are assigned from information on the upper layers, for example, the UDP port number. QoS provisioning is subject to the individual network nodes, like routers, switches, and bridges. Table 4.1 summarizes these classes. Although classes for audio and video exist and cover precise constraints for delay and jitter, it is neither guaranteed that their constraints can be met, nor is a reliability constraint defined, so packet loss is not covered by this definition. Meeting the requirements is solely dependent on the network infrastructure, which means the fact that the underlying network has enough capacity to provide a service quality that lies within these specifications. To reserve network capacity, applications can claim resources for streams using the Multiple Stream Reservation Protocol (MSRP). Network resources, and thus, QoS, can then be guaranteed if all devices in the network are obeying the 802.1Q standard. In this form, the bridged network can guarantee low latency (although with no specific upper bound) and the absence of congestion loss.

An important feature for TSN in terms of both reliability and low-latency is Frame Replication and Elimination for Reliability (FRER), as defined in 802.1CB [80]. By this means, streams are replicated into several sub-streams that are routed along the different paths throughout the network. The sink gains the ability to recreate lost packets through this redundancy. Replications can be identified by the packet ID, and the order can be restored at the sink. Using this technique highly increases the reliability of streams by the cost of higher network load. In contrast, latency is reduced by eliminating the necessity of retransmissions.

Another notable standard is the 802.1CM [81] that specifies the usage of TSN for Fronthaul in mobile telecommunication systems. IEEE 802.1 TSN concepts are also naturally interoperable with all IEEE protocols like IEEE 802.11, IEEE 802.15.1 and IEEE 802.15.4.

Table 4.1: IEEE 802.1Q Traffic Classes [82].

Priority	Traffic class
0	Background
1	Best effort
2	Excellent effort
3	Critical application
4	Video, latency and jitter <100 ms
5	Voice, latency and jitter <10 ms
6	Internetwork control
7	Control data traffic

4.4.2 IETF DetNets

In cooperation with the IEEE, also the IETF pushes standardization efforts for URLLC, and has established the DetNet Working Group [83]. The IETF concentrates on the Network Layer and is not limited to 802.1-based networks. DetNets concentrate on both IP-based networks and label switched networks [84] and can perspectively be incorporated in Autonomous Systems, although those groups are not specified as target systems of DetNets. The goal of the DetNet WG is the standardization of protocols for the establishment of multi-hop paths for flows with particular requirements in throughput, timing, jitter, and ultra-low packet loss [85]. It supports applications with critical timing and reliability constraints with means of time synchronization in the range of microseconds, latency guarantees, guaranteed packet loss ratio, high bandwidth, controlled congestion behavior, resource reservation, and robustness. These criteria are often summarized by the two keywords *deterministic latency* and *zero loss*.

DetNets are built for certain use cases [86]. One of the use cases is the application as fronthaul and backhaul networks in telecommunication networks to support URLLC services. Here, the DetNet serves as an IP-based replacement for the proprietary point-to-point protocols that are in use today, and can act as a means to establish network slicing within the Access Networks. But also wireless networks are subject to the DetNet standard. Similar to other IETF standards, DetNet is agnostic to the used Layer 2 and Layer 1 schemes.

DetNets specify mechanics for resource allocation and deterministic routing on the IP layer, as well as flow replication on the transport layer [87]. A *flow* is the basic service unit in a DetNet, which is a registered relation between endpoints with annotated QoS requirements [88]. Flows coexist with normal traffic in the same network. A resource allocation scheme for flows ensures that routers prioritize DetNet flows over normal traffic, and enforce traffic shaping, scheduling, and queueing disciplines. For service protection, a DetNet can drop misbehaving flows, routers, or applications for out-of-order packet transmissions. Flow replication over multiple paths can help to mitigate flow failures due to such events. A central controller, the Path Computation Element (PCE), is responsible for flow registration and control. The Resource Reservation Protocol (RSVP) [89] can be used as a management scheme.

DetNets are part of the IP Layer, so they build on top of any Layer 2 protocol. They are capable ob multi-layering, such that a DetNet can be built on top of another DetNet subnetwork, as shown in Figure 4.5. The IP Layer is divided into Service and Forwarding sub-layers. The Forwarding sub-layer is responsible for implementing the regimes that are in place in order to enforce determinism within the flows, whereas the Service sub-layer is responsible for flow management. The end-to-end service of a DetNet is defined between the application layers. Specific Edge Nodes and Relay Nodes can be placed to forward DetNet traffic. Edge Nodes are interfaces to other IP-based protocols, such as TSN, while Relay nodes are DetNet-internal routers. Moreover, a Transit Node is a DetNet-aware router that only implements the forwarding sub-layer.

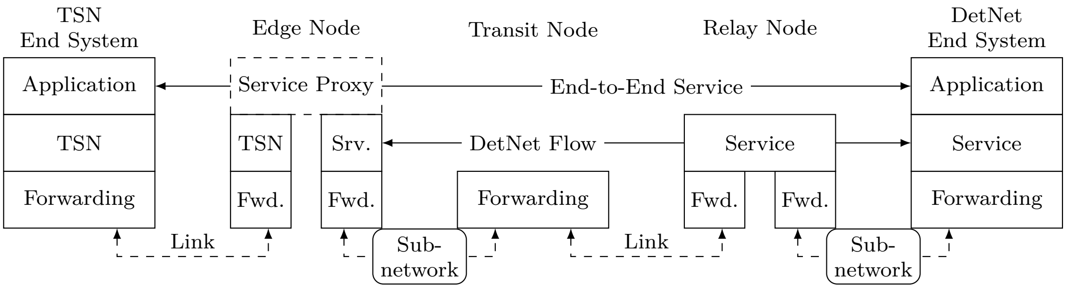
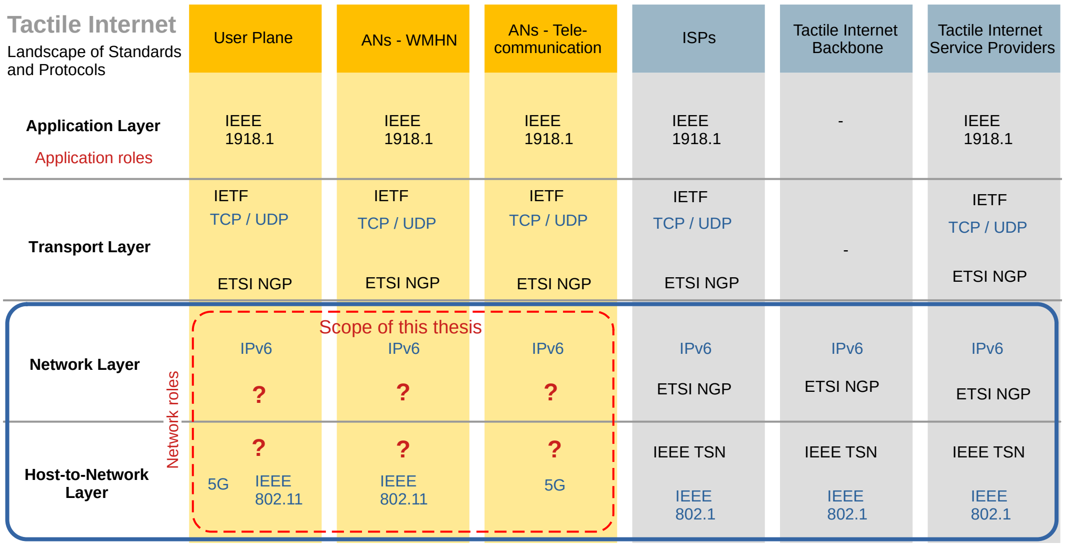


Figure 4.5: Overview over DetNet sublayers, adapted from [88].



Tactile Internet Landscape of Standards and Protocols	User Plane	ANs - WMHN	ANs - Tele-communication	ISPs	Tactile Internet Backbone	Tactile Internet Service Providers
Application Layer Application roles	IEEE 1918.1	IEEE 1918.1	IEEE 1918.1	IEEE 1918.1	-	IEEE 1918.1
Transport Layer	IETF TCP / UDP ETSI NGP	IETF TCP / UDP ETSI NGP	IETF TCP / UDP ETSI NGP	IETF TCP / UDP ETSI NGP	-	IETF TCP / UDP ETSI NGP
Network Layer	IPv6 ?	IPv6 ?	IPv6 ?	IPv6 ETSI NGP	IPv6 ETSI NGP	IPv6 ETSI NGP
Host-to-Network Layer	? 5G IEEE 802.11	? IEEE 802.11	? 5G	IEEE TSN IEEE 802.1	IEEE TSN IEEE 802.1	IEEE TSN IEEE 802.1

Figure 4.6: Overview of the Tactile Internet standards landscape.

The DetNet standard is an important prerequisite for the further deployment of the Tactile Internet, as it is capable of enforcing QoS-requirements for applications.

4.4.3 ETSI NGPs

European Telecommunications Standards Institute (ETSI) has concluded a work item on the IPv6-based Tactile Internet in 2017 [90], but no corresponding standards have been released yet. In a 2018 Group Report from the ETSI Next Generation Protocols (NGP) group [91], recommendations for new transport technologies are proposed. The idea is to enhance the IP layer to introduce QoS management on a control plane on each Internet router to overcome the limitations of host-only transport layer solutions. By using IP-based in-band signaling, routers can identify packets and their flow properties to ensure requirements. Hosts are allowed to probe the network path and collect information from the affected routers. The method can be used either for static initial route setup and QoS provisioning, or dynamically to re-establish an already provisioned route in case of link failures, re-routing, or other changes in the network or within the application. Since the routers are aware of the coming data and its requirements, this method allows for giving a certain service level guarantee from within the network. It can be implemented within the TCP handshake, for example.

4.5 Discussion of the Existing Approaches

The existing standardization approaches all focus on URLLC, which is the key element of Haptic Communication. Only the IEEE 1918.1 approach goes a step further in that it defines a complete Tactile Internet infrastructure in a top-down style. Therefore, it is ahead of the other standards in this regard. However, it will rely on other standards when it comes to details of the actual network infrastructure. Arguably, the IEEE 1918.1 standards are more related to the Application Layer, while the other approaches target lower layers. When considering the areas of interest for the individual standards, they complete each other very well. With Figure 4.6, we prospect the future potential application of each of these standards on the communication layers and in different parts of the Internet. The prediction is vague in the sense that, for example, TSNs are not designed to fit the role of a Host-to-Network-Layer protocol for the Internet core. However, similar or adapted protocols may evolve to establish URLLC in these parts, too. The Access Layer parts are highlighted since these will likely solve the 1 ms-problem in the traditional producer-consumer pattern, as exposed in our introductory telesurgery scenario. But here, on the lower layers, the standardization approaches do not fit.

NGPs and TSNs are wired protocols, so they are not suited for wireless Access Networks. The only candidates that offer URLLC are telecommunication networks (beginning from 5G), but several open questions remain for these as the specifications are ongoing. Especially the design of URLLC-capable wireless multi-hop networks is the subject of continuous research. The wireless medium still challenges the development of protocols, and the haptic modality is not yet established in current wireless networks.

We believe that the organization of the IEEE 1918.1 standardization process into the development of several loosely coupled sub-standards is a good approach, since yet too many unknowns exist. The first sub-standard, IEEE 1918.1.1, covering haptic coding, covers a core aspect of Haptic Communication as addressed in current literature. Coding is also a very confined and general aspect, which makes it suitable for early standardization. However, the top-down approach of the IEEE 1918.1 standardization process may be suboptimal with regards to leveraging the current state of Internet technology. The draft Application Layer protocol is intended to be agnostic to the used Transport- and Network Layer protocols, although it is unlikely the IP-based networks will be replaced any time soon. Packet-switched networks in combination with IPv6 have now been established as a technology, and the development is unlikely to wind back to circuit-switched alternatives, despite they might offer better control over service quality. The Tactile Internet must, therefore, be thought from the bottom up.

With the three-fold contribution of this thesis, we primarily address the gap on the lower layers of Access Networks. Our latency model and the Tactile Coordination Function close gaps both in the allocation of network resources and in QoS enforcement on these layers. Our application framework, HDTF, covers aspects from the Application Layer, so that both application and network roles are addressed with this thesis.

CHAPTER 5

Architecture Proposal

In this chapter, we sketch a software architecture for the Tactile Internet. Our proposal assumes Digital Twins as a solution to the 1 ms-problem and, otherwise, follows a bottom-up approach, consolidating the established Internet standards and open protocols that are available and related to the URLLC use case. It is thus a thin layer on top of the current Internet architecture. We include the TSE, as proposed by IEEE 1918.1 as an Edge Cloud element and use it to deploy the Digital Twins. TSEs can be placed within the current Internet architecture in order to transition to a Tactile Internet, when the Access Network models that we develop in later chapters can be employed. Finally, we identify core conceptual ideas that we address in Parts III and IV of this thesis in more detail.

5.1 A Haptic Digital Twin Framework

To cover the elaborated open problems, we propose a Haptic Digital Twin Framework (HDTF) for the Tactile Internet. The purpose of the HDTF is to untangle and solidify the relations between both network and application roles as motivated in the forgoing chapter. Figure 5.1 shows an overview, where the 1 ms-problem is solved by independent and interchangeable network and application roles. The overview reflects the separation of the roles, as well as their equal importance to the application. Traditionally, a “network role” would simply reflect the protocol stack used, probably including Application Layer communication protocols, of a networked application. In the overview, however, we must distinguish at least between the wired Internet core (Backbone), and an Access Network that is capable of URLLC. The real endpoints in the overview are located at the end of the communication lines. In our initial telesurgery example, these would reflect the surgeon and the patient. The Digital Twins would be placed at the TSEs at the edges between Access Networks and the Internet Backbone. The Internet Backbone is responsible for the synchronization of the TSEs.

On top of the Operating System, the HDTF cares for a unified interface to the application with its Digital Twin roles. Both roles can be addressed in the same manner and using the same interfaces, which results in high reusability of code. the HDTF connects all endpoints of an application through the network, so HDTF is a true framework rather than just a library of functions. A set of management subsystems are required to connect the application endpoints and monitor and control both the Quality of Service (network metrics) and the Quality of Experience (application metrics).

5.1.1 QoS and QoE Model

As both latency and reliability are essential requirements, the appropriate QoS and QoE models are an integral part of the architecture. The QoE is application-specific and may not be defined in terms of any network metric, but must be ultimately mapped to QoS metrics suited for the used network role. The HDTF must provide respective mappings for common

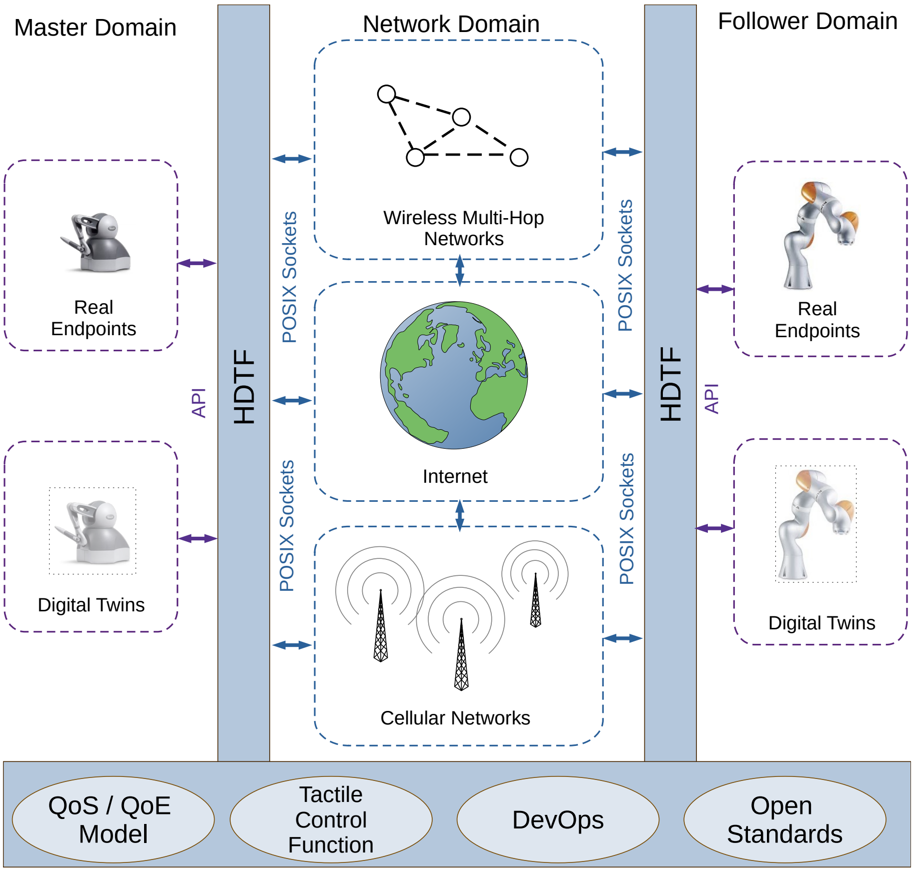


Figure 5.1: Proposed Framework Architecture.

application metrics, such as the accuracy or precision of a physical movement, or similar. These metrics must be defined both in the physical and the virtual part of a Digital Twin.

The QoS requirements can be formulated as, e.g., “The Access Network must provide x ms latency between the endpoint and the Digital Twin” or “The virtual and the real part of a Digital Twin require a constant throughput of y bps.” The formulation of QoS requirements results in a classic consumer-producer view on the relation between network and application, which means that the physical limitations must be met. The framework must support such measures, which can be part of the resource allocation subsystem.

The model requires the proper assessment of the capabilities of the given network resources. Special care must be given for Access Networks that have to host both TSEs and application endpoints. Especially for wireless multi-hop networks, the capability modeling regarding URLLC is complex. The literature has not tackled the modeling of latency in WiFi multi-hop networks sufficiently yet. This modeling aspect is thus our main concern in this thesis, which we cover in detail in Chapters 6, 7, and 8.

5.1.2 Tactile Coordination Function

The given QoS constraints, after modeling and planning, must be met in terms of allocating resources within the network. For the Tactile Internet, two aspects are especially important. First, the allocation of new computational resources (e.g., TSEs) must be organized. Second, the communication resources must be allocated on all protocol levels. Communication must be organized by means of communication *flows*, which are primitives that organize the QoS requirements and allocated resources, such as bandwidth, medium time, network routes, etc.

As a first measure, a new communication class for Haptic Communication, similar to audio and video classes, should be introduced. Traffic shaping mechanisms must enforce the higher priority for Haptic Communication both on the router level (Network Layer) and on the (wireless) medium (Link Layer). Techniques exist for many network infrastructures. But

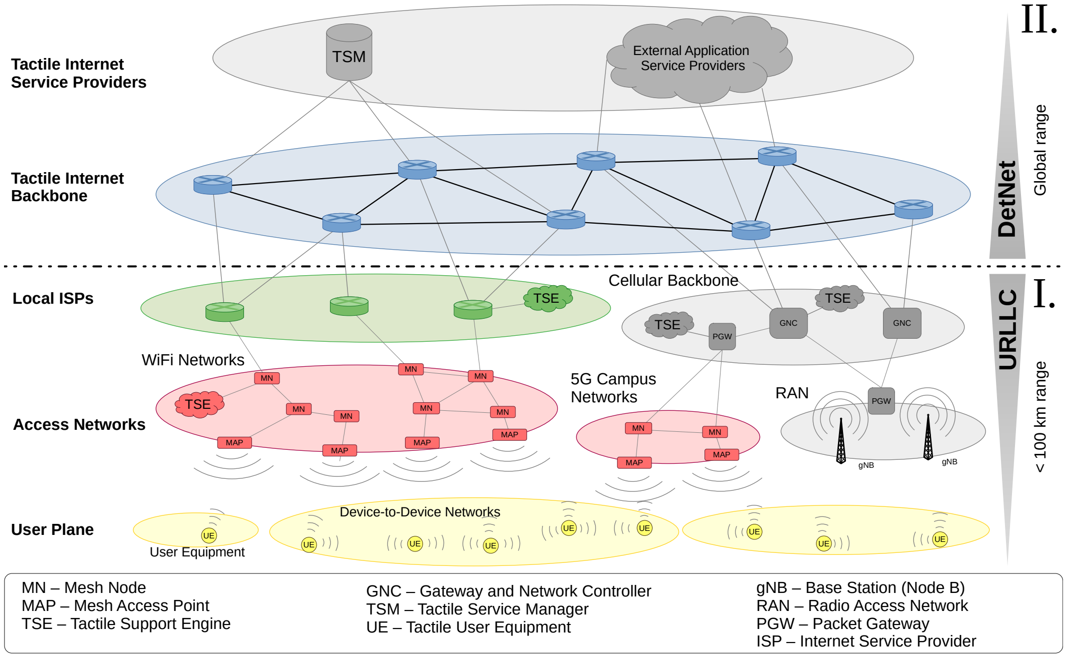

II.
Tactile Internet Service Providers
TSM
External Application Service Providers
Global range
DetNet
Tactile Internet Backbone
I.
URLLC
Local ISPs
TSE
Cellular Backbone
TSE
PGW
GNC
TSE
GNC
WiFi Networks
5G Campus Networks
RAN
PGW
Access Networks
TSE
MN
MAP
gNB
gNB
< 100 km range
User Plane
UE
User Equipment
Device-to-Device Networks
MN – Mesh Node
MAP – Mesh Access Point
TSE – Tactile Support Engine
GNC – Gateway and Network Controller
TSM – Tactile Service Manager
UE – Tactile User Equipment
gNB – Base Station (Node B)
RAN – Radio Access Network
PGW – Packet Gateway
ISP – Internet Service Provider

Figure 5.2: Architecture of the Tactile Internet. Specifications for Tactile Internet Service Providers and Mobile Telecommunication Networks are credit to the IEEE 1918.1 workgroup [5].

again, wireless multi-hop networks impose problems in this regard, as on the shared medium prioritization is hard to enforce. We address this problem for WiFi multi-hop networks in Chapter 9.

5.1.3 DevOps

With a PaaS approach, capacities can easily be allocated and deallocated with a fixed set of interfaces and software components in order to minimize the amount of hardware-specific code. The platform must provide a clean interface to Digital Twins that allow for state updates and synchronization of the real and virtual spaces. In Chapter 10, we detail the HDTF and implement a subset of this platform in form of a Haptic Application testbed.

5.1.4 Open Protocols

The Internet has grown over years through open approaches provided and improved by a large community. URLLC will be a future core functionality, especially with 5G and the development towards 6G. The first Tactile Internet Applications may simply rely on single-hop 5G network hardware operating in URLLC mode, which can be regarded as a relatively homogeneous system under the control of a single authority. But in the future, service integration between different ISPs, different vendors, and different technologies will be necessary to provide Tactile Internet Services to the broad public. A robust Tactile Internet architecture must, therefore, integrate existing, even legacy hardware.

5.2 Embedding within the Internet Architecture

Preserving the current structure as far as possible and adding the components pointed out, a structure as is depicted in Figure 5.2 can be drawn. It is also partly based on the architecture model provided by the IEEE 1918.1 Working Group [5].

Access Networks of local ISPs are in charge to host the communication services required for Haptic Applications. URLLC is the main type of communication within these networks, as far as Tactile Internet Applications are considered. As there is no distinction between application modes, Haptic Communication coexists with non-URLLC services in the same networks. It is subject to future network solutions to provide the proper network slicing or some prioritization means to separate the URLLC applications from less demanding traffic in terms of QoS. (For multi-hop WiFi networks, for example, we propose such a prioritization method in Chapter 9.)

TSEs are concentrated mainly within the Access Network Plane. They can be, however, part of the different networks, either that of local ISPs, or of local network infrastructures under the control of end-users (Companies, Campus Networks, etc.). Local installations are preferable over Edge Cloud solutions in professional use cases, where the application endpoints are part of some corporate network infrastructure. But they can be offered as well in the form of a PaaS, as it is common today for example for cloud computing solutions. In this case, TSEs will be part of the infrastructure of some local ISP.

The Core Network Plane (II.) is then subject to global ISPs and their Autonomous Systems and Exchanges. The provision of deterministic latency and zero-loss data communication across the Backbone is a key concern here, and IP-based DetNet protocols can be in charge to provide the proper service quality. According to the IEEE 1918.1 approach, special services for the Tactile Internet can also be part of the Core Network Plane, for example, Tactile Service Managers (TSMs), and service providers for TI Application Servers. A TSM is responsible for haptic flow registrations, and controls the QoS requirements of applications from end-to-end.

5.2.1 Tactile Support Engines

Tactile Support Engines are Edge Cloud entities dedicated to the Tactile Internet that solve the 1 ms problem. They host application services close to haptic endpoints and thus allow for the provision of strict latency requirements even for intercontinental applications. As the range of TSEs is limited to 100–150 km, they must be placed relatively low in the Internet hierarchy. They are best located within Access Networks, as the distance to the respective user equipment should be as small as possible.

TSEs are specified as part of IEEE 1918.1, but as the complexity of applications can not be foreseen due to the few existing, mostly experimental use-cases, the current state of standardization can not make any approach to a concrete service definition yet. The role of a TSE, however, is mainly that of a Digital Twin which replaces the communication to the endpoint by a digital proxy that can be accessed faster. We pursue this vision further in the next chapters, where we formalize a Tactile Internet Application model.

5.3 Summary

In this chapter, we have drawn an architecture sketch of both the Tactile Internet itself and the structure of its applications.

The following Chapter 6 formalizes all our previously made statements. It contains a formal definition of both Haptic Applications and Tactile Internet Applications, as well as a formal basis for latency and reliability modeling for Access Networks.

In Part III of this thesis, we address the problems on the Access Layer. The QoS and QoE modeling for Access Networks is addressed in Chapters 7-8. Chapter 9 is concerned with the Network Resource Allocation part specifically for WiFi multi-hop networks, which lack a dedicated service class for Haptic Communication.

An approach to conceptualize a Haptic Application Layer is made in Part IV. In Chapter 10 we introduce our concept of a Haptic Digital Twin Framework (HDTF) in more detail and present a partial implementation.

CHAPTER 6

Formalization

In this chapter, we give a formalization of both the application model and the network model. Furthermore, we give an approach for modeling latencies in multi-hop wireless Access Networks. For analysis, we outline an axiomatic system for a probabilistic model that describes both the latency and reliability distribution within a wireless Access Network. The chapter ends with an outlook on the queueing problem within wireless Access Networks.

6.1 Application Model

We define the Tactile Internet Application as the core element of concern to begin our argumentation. First, we introduce the term Haptic Application, a similar concept but differs by its locality property.

6.1.1 Definition of a Haptic Application

A *Haptic Application* is defined by two properties.

a. Its endpoints exchange haptic data as described in Section 2.2.

b. The specific QoS or QoE metrics presented in Sections 2.3 and 2.4 apply to its data flows.

We do not make any assumptions about the Application Layer, e.g., what the user interface looks. We also do not assume a human user is involved. The individual data content does not matter, so we only focus on the non-functional parameters of the respective communication flows. As a general category of traffic constraints, we assume 1 ms round-trip latency, 99.999 % reliability in terms of mean successful packet transmission rate, and a packet rate of 1 kHz, as it has been intended with URLLC in IMT-2020. This set of constraints covers most of the use cases from Section 2.3.

Haptic Applications can mitigate problems with propagation delay by keeping a limited distance between the endpoints (EPs). Classic control systems, Networked Control Systems, teleoperation systems, and Augmented Reality Systems that have tight latency bounds in the range of milliseconds are therefore limited in their communication distance. Haptic Applications are the classic use case for URLLC services, as they are used in industry and control systems. For the sake of simplicity, we further assume that Haptic Applications are unicast applications. If an n-to-m relation is required between EPs, a peer-to-peer or a client-server relation can be established as pairwise unicast connections without loss of generality. Each peer or server is considered an EP. Multicast or broadcast connections are not subject to our consideration and may be subject to further research work.

6.1.2 Definition of a Tactile Internet Application

A *Tactile Internet Application* is defined by two properties.

a. It is a Haptic Application.

b. Its endpoints are distributed worldwide over the Internet.

A Tactile Internet Application is, as a Haptic Application, defined by the shape and quality metrics of its network traffic. However, Tactile Internet Applications can span intercontinental distances and use the Internet as a transport medium. Tactile Internet Applications can thus not avoid high propagation delays between their endpoints.

A Tactile Internet Application must at least have the structure as visualized in Figure 6.1. It is divided into endpoints (EPs) and Digital Twins (DTs) that communicate over the Internet Backbone. The role of a DT is to reduce the virtual distance between EPs that are effectively too far away from each other to maintain the required communication latency. We, therefore, assume DTs reside in servers within Access Network range of the associated EPs, as the 1 msdistance around EPs is almost certainly within the coverage area of the respective Access Network.

Planning TSE locations within Access Networks also has the advantage of network homogeneity. Access Networks are often controlled by a single provider, which eases the adoption of URLLC protocols. Enforcing URLLC connectivity across provider boundaries, as would be necessary if the TSE was located outside of the respective Access Network, is harder to achieve and requires the support of the adjacent ISP.

We follow here the notation of the IEEE 1918.1 working group and categorize the EPs into master and follower EPs [5]. This distinction is according to the application of haptic teleoperation, where a master EP controls a follower EP, thus imposing a direction of communication. The same assumption is made for NCSs. They are divided into a controller and a plant and, therefore, also have an implicit communication direction. However, each of these relations is bi-directional, as control data and feedback are always required.

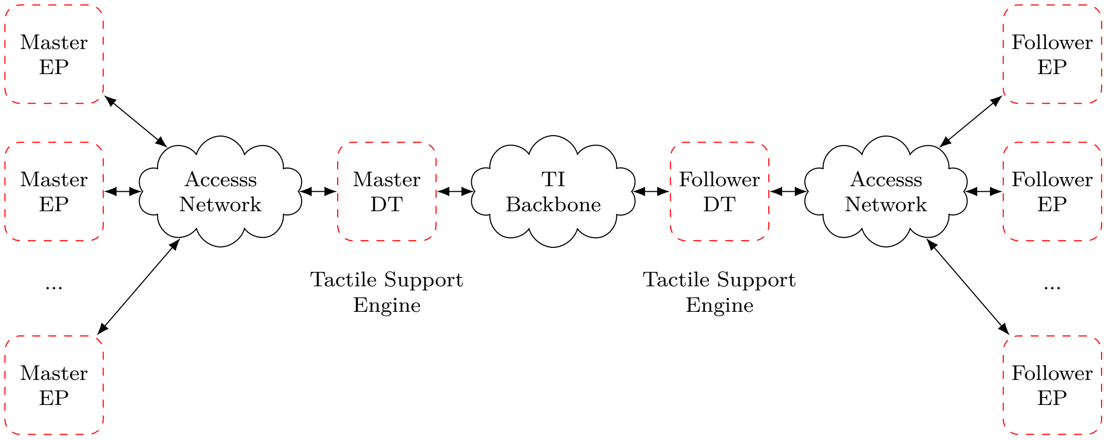


Figure 6.1: Tactile Internet Application model.

As Backbone and Access Networks serve separate roles within this model, they have different requirements. The distance between an EP and its closest DT is subject to the same QoS and QoE constraints as a typical Haptic Application. The propagation delay between these hosts must be minimized, which can be achieved by distance reduction. In addition, the Access Network must obey all required constraints in terms of latency and reliability. In other words, ANs need to implement URLLC in order to support TI Applications.

The Backbone is responsible for DT synchronization and must support the data and information flows described in section 4.2.1. The data that needs to be exchanged, which may, for example, consist of 3D-models, surface information, or kinesthetic models, differs from the haptic data that is exchanged between EPs and DTs. A major difference is that this data can be updated incrementally, as the objects (and environments) change gradually over time, and changes can be predicted by interpreting the user's interactions. In the optimal case, DT synchronization requires only the continuous correction of prediction errors. From the Backbone, only a deterministic temporal behavior is required, so that a fixed deadline for synchronization can be guaranteed. This time span is, at the same time, the maximum age that a DT has in terms of actuality. Synchronization errors that are known to be of a maximum age can be handled by the user, e.g., by interrupting an operation for a short time period to wait for proper synchronization of the system.

6.1.3 Example Use Case: Teleoperation of a Robot

Consider a teleoperation system as depicted in Figure 6.2, where a human operator controls a robot arm. A similar use case has been investigated by Xu et al. [92], where they introduced model mediated teleoperation as a solution that can enable long-distance operation. Both EPs maintain local control loops on their system states to ensure a stable local operation. The human operator at the master EP interacts with a haptic input device that allows for precise control of the teleoperator's setpoint. The force feedback on the haptic input device also stimulates the operator to regulate the movement commands according to the remote EPs system state. On the follower EP, the teleoperator has to maintain its state as well and move towards the given setpoint.

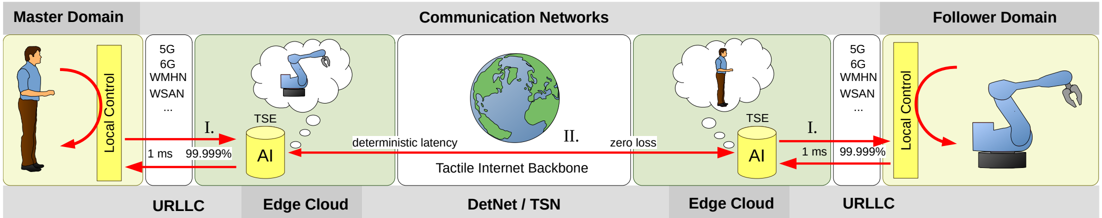


Figure 6.2: Example use case of a haptic teleoperation system as a Tactile Internet Application. The Access Network Plane is implemented by URLLC technologies (I), and the Core Network Plane (II) uses DetNets for longer range communication. The perceived latency is solely defined by the Access Network Plane, as long as the system is in steady state, i.e., the Digital Twins are synchronized.

To maintain a 1 ms response time over a long-distance communication link with large propagation delay, the master EP creates a Digital Twin of the teleoperator and its environment. The environment model is created by a 3D scanner as a 3D point cloud, where a 3D mesh can be extracted. changes in the environment are discovered by the 3D scanner, and the respective 3D model is updated at the master DT. The human operator thus interacts with the teleoperator's virtual copy, to which it can maintain the required response time. Similarly, the operator's virtual twin is migrated towards the teleoperator to model the responses of the human to changes in the teleoperator's environment.

The URLLC in the Access Network between the EPs and DTs can be established with any network technology capable of the required QoS constraints. This can be either 5G, 6G, WiFi, or any wired connection. The Backbone, which has less strict requirements, a DetNet solution may be appropriate to control certain delay sources, as buffer bloat, for example.

Although the DTs are not an accurate representation of the respective remote EP and their environments, the application can offer a low perceived latency. Due to delayed updates of the DTs through the Backbone, a certain residual error in the time domain can occur. Each change in the follower's environment must be synchronized with the master DT, which must per definition include end-to-end delay. Finally, the follower state and the master DT can differ so much that the system has to be considered as *out of sync*, such that it must be paused in order to re-establish synchronization. However, during in-sync operation, such a system can offer a perceived latency that is lower than the actual, physical latency between the endpoints.

6.1.4 End-to-End vs. Perceived Behavior of Tactile Internet Applications

The architecture of TI Applications implies some effects on the end-to-end latency behavior. The end-to-end latency between master and follower is still defined as the amount of time that information takes to be shared from one EP to another. The sharing now happens in three distinct steps across the boundaries of the respective Access and Core Network Planes, as is depicted in Figure 6.3.

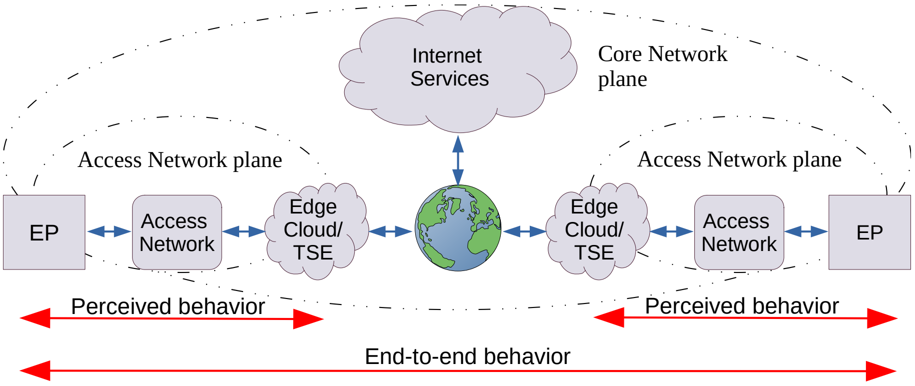


Figure 6.3: Proposed two-tier structure of Tactile Internet Applications.

We must, therefore, introduce a distinction between the physical *end-to-end latency* and the *perceived latency*. The perceived latency is the latency between the EP and its nearest DT in the Access Network. This latency can be significantly smaller than the end-to-end latency, which includes the propagation delay, while the perceived latency does not. The perceived latency only depends on the distance between the EP and its peer DT. With this application layout, it is possible to build TI Applications on a worldwide scale, as the perceived latency can be controlled by the distance between EP and DT.

6.1.5 Application States

In order to maintain low perceived latency, the EPs and DTs have to be synchronized, i.e., information that has to be referenced by an EP must be available and up-to-date at the respective peer DT.

The perceived latency can be observed only when the TSE can offer a full user experience without the necessity to synchronize with the other endpoint of the application. We call this phase the steady-state in the tradition a statistical process modeling, as it usually occurs after some time of operation. Here the endpoints can fully interact with their local TSEs that have enough information at hand to serve the full application spectrum. When this is not the case, for example when the user leaves the defined application space, or the remote site changes unforeseen, then the TSEs must synchronize again with the remote site. During this transition phase, the system must be either pause its operation, or it has to cope with the much larger end-to-end between the EPs. Figure 6.4 shows the corresponding state diagram.

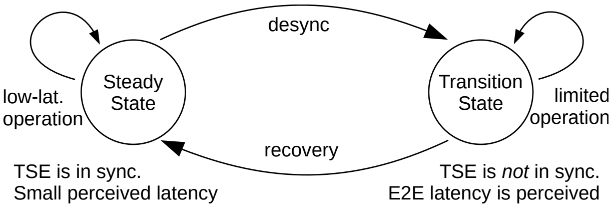

desync
Steady State
Transition State
low-lat. operation
limited operation
recovery
TSE is in sync. Small perceived latency
TSE is *not* in sync. E2E latency is perceived

Figure 6.4: State diagram of a Tactile Internet application.

This system behavior introduces two additional quantities to measure as a means of the application Quality of Experience that we define as follows. First, the proportion that the system remains in its steady state should be maximized in order to perceive the minimum possible latency. We call this the *steady state share* t_{SSS}:

$$t_{\mathrm{SSS}} = \frac{\text{time spent in steady state}}{\text{operational time}}$$

It is desirable to reach a 100 % share, but this is limited due to several factors, e.g., the computational capacity of the TSE, the bandwidth of the global link, the complexity of the remote site and its environment, and on the user's interaction as well. Second, when in the transition state, the system must make make effort to recover and synchronize again. The *transition time* t_{trans}, i.e. the time it needs for recovery, depends on the global link bandwidth and the severity of the desynchronization. It is the time that is needed to re-synchronize models on the TSEs.

6.2 General Network Model

The function of a communication link is, in general, subject to complex interactions between heterogeneous systems, most of which behave non-transparent. The reference models for the Internet, however, grant some means of abstraction. All nodes on the Internet operate after the store-and-forward principle. Packets have to be received completely before they are decoded and forwarded to the next intermediate node. It is thus an intrinsic property of Internet communication that with every *hop* a certain amount of time elapses, so the end-to-end delay correlates with the number of hops from source to destination. The store-and-forward operation also introduces *queueing delay*, as packets are transmitted one after another.

A common modeling approach is graph-based modeling, where *nodes* (routers, switches, hosts, access points, etc.) are interconnected by communication *links* (optical fibre, twisted-pair copper wire, wireless transmission, etc.). Although effective for small-scale networks, that approach would not be powerful enough at an intercontinental scale. For an end-to-end latency or reliability analysis, we need to consider – at least – all the network nodes and all links between them on the path from the source to the destination host (i.e., the flow path), and model their behavior. Unless the nodes on a flow path belong to one specific network application, they have to allocate and share their resources (i.e., the transmission capacity of the links and switching capacity of the routers) with other network applications. A complete model must include all applications, nodes, and links on the Internet to be completely accurate.

As we have elaborated already, the Internet Backbone consists of a network of Autonomous Systems (ASs) that operate relatively independently of each other. Access Networks organize the interconnection of the end-user devices, which leads to the following view illustrated in Figure 6.5. The interconnected ASs in the Backbone are almost entirely comprised of wired networks. These are well-known and their behavior can be modeled as a chain of queues, where routers are operated redundantly in a complex interconnected structure (Barakat et al. [93] have shown various modeling techniques). For our application model,

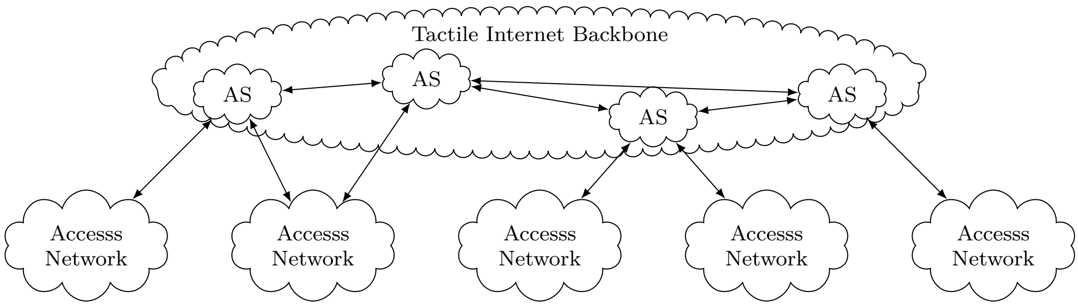


Figure 6.5: Interaction between network Planes.

the time constraints in the Backbone are relatively relaxed, as latency can be compensated by the AI within the Edge Cloud, and determinism is a more important aspect. Within the Access Networks, timing constraints are strict, but quantities are manageable and are likely homogeneous. They are also often managed by a single authority. Due to the necessary Edge Cloud interface between Access Networks and the Backbone, it is sufficient to consider only one of these networks at a time rather than modeling the whole system from end to end.

The abstract view allows for modeling the networks independently from one another, for several reasons. First, latencies behave additive and reliability behaves multiplicative, so the respective end-to-end metrics can be calculated by cumulating values of the affected sub-networks. Second, the Edge Cloud subsystems of Haptic Applications act as a natural boundary, as the different network planes (Core Network and Access Network) are independent during steady-state operation. It is often only interesting to model the behavior of a specific Access Network, but not that of the overall system.

In the mid-term future, most Access Networks may eventually have mostly wireless communication links. We can model them using the wireless multi-hop network (WMHN) abstraction, which assumes a set of wireless nodes that are connected in a mesh topology. This is true for the two most prominent candidate technologies for Access Networks: WiFi networks and cellular communication networks. Their infrastructure, as derived from Chapter 5, for supporting Haptic Applications can be visualized as in Figure 6.6, where the routers and the TSE are part of the infrastructure provided by the Internet Service Provider (ISP). An ISP may be able to control all network resources associated with an Access Network (AN) in a centralized manner, which includes the routers on a system level, and the radio channels that their Physical Layer hardware operates on. This is the case for current (5G) and future cellular networks (6G and beyond). But also WiFi networks constantly gain more resources in the form of ISM frequencies, such that adjacent ANs can operate on distinct radio frequencies. The ISP can then model portions, or even an entire AN in full detail in a graph-based model to configure load behavior, network coverage, resource allocation, load balancing, latency, and reliability modeling.

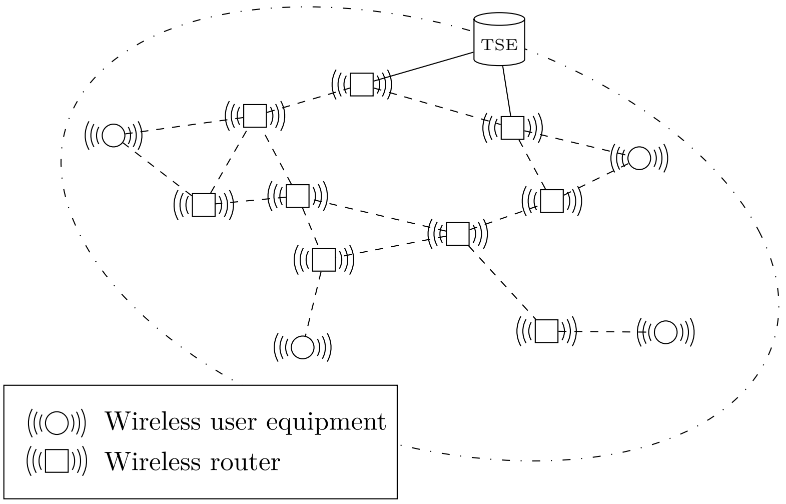


Figure 6.6: Confined view of a wireless Access Network with bounded coverage area.

It is necessary to state that the model is simplified in many ways. The TSE may not necessarily act as a border device between AN and the Backbone, and there can be more than one within a single AN, whether these instances are all replications of the same system or whether they share the network's workload among each other. There may also exist wired connections between nodes. However, for the sake of simplicity, we neglect these circumstances in our modeling. Our goal is to obtain a modeling technique for latency and reliability in the AN model provided in Figure 6.6 in order to provide URLLC services, which requires us to make certain generalizations and simplifications.

6.3 Haptic Flow Model

For the communication between endpoints and Digital Twins in a Haptic Application, the *flow* abstraction is useful for traffic shaping and modeling. A flow is an agreement between endpoints about QoS requirements. We define the term haptic flow as follows.

A *haptic flow* is a unicast relation between two nodes $s, d \in \mathcal{V}$, where s is the source host and d is the destination host. A flow is annotated with QoS and QoE constraints that have to be met. Flows are realized as a *flow path* that represents the set of links that the data is forwarded along.

In any given AS (either an AS in the Internet Backbone or an AN), we consider the set of flows individually. We denote the number of flows in an AS as n_f. Each flow j, $1 \leq j \leq n_\mathrm{f}$, is realized on a path $\mathcal{P} = \{(v_0, v_2), (v_1, v_2), (v_2, v_3), ..., (v_{l_j-1}, v_{l_j})\}$, $\mathcal{P}_j \subset V$, where v_0 is the source node and v_{l_j} is the destination node. l_j is the hop count of path $\mathcal{P}_j$. The following Figure 6.7 shows such a flow within an example network.

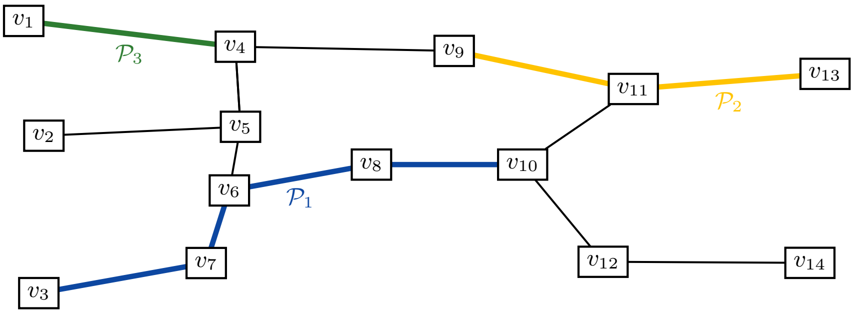


Figure 6.7: Example of $n_\mathrm{f} = 3$ flows in a network of $n_\mathcal{V} = 14$ nodes. Flow 1 in blue on path $\mathcal{P}_1 = \{(v_3, v_7), (v_7, v_6), (v_6, v_8), (v_8, v_{10})\}$ has four hops ($l_1 = 4$), flow 2 in yellow on $\mathcal{P}_2 = \{(v_9, v_{11}), (v_{11}, v_{13})\}$ two ($l_2 = 2$), and flow 3 in green on $\mathcal{P}_3 = \{(v_1, v_4)\}$ has 1 hop ($l_3 = 1$).

The flow specification organizes information on QoS and QoE requirements. Such a specification can be a set of constraints, as it is specified for example by RFC 2210 and RFC 2211 [34, 35] Integrated Services. Table 6.1 shows the minimum set of parameters that are required, together with typical values drawn from our literature review in Section 2.3.

Table 6.1: Parameter set for specifying a haptic flow.

Parameter	Symbol	Range		
		min.	typ.	max.
Message rate	r	50 Hz	100 Hz	1000 Hz
Payload size	s	4 B	24 B	3000 B
Max. latency	t_max	0.5 ms	1 ms	100 ms
Reliability	R	$1-10^{-2}$	$1-10^{-5}$	$1-10^{-9}$

For QoS provision, an admission control system for haptic flows must be put into place. It keeps a flow register, such that every flow within a AS is known and registered. It needs to keep information such as the flow specification, the path through the network, possibly among other parameters. These registrations are required, as we have to ensure that the network load can be assessed as gross value (i.e., including all overhead down to the Physical Layer). As we will see in our further elaborations, especially in Chapter 8, we need such exact information on the level of the entire AN to assess the latency distributions of the haptic flows. Therefore, our approach requires the Integrated Services approach, and a reservation protocol such as RSVP [34] must be utilized.

Since the modeling can be restricted to within the AN or AS boundaries, the parameters in Table 6.1 are meant to be expressed as requirements that have to be met within the specific network segment. So, the values are not to be interpreted as e2e values but have to be broken down into the portion of the specific AS or AN.

6.4 Latency and Reliability in Multi-Hop Networks

Reliability can be viewed as a sub-aspect of packet latency since packet losses can be interpreted as packets that were exposed to infinite communication delay. When reliability is defined as the packet error rate, the nominal reliability is the number of packets that were exposed to infinite delay, divided by the overall number of packets sent. Latency is thus a generalization of reliability, and we can reduce our modeling to just packet latency.

In Chapter 2, we have already pointed out the main sources of packet delay: propagation delay, queueing delay, arbitration delay, propagation delay, transmission delay, and acknowledgment delay. We can now derive a more formal definition and the interplay between these. Assuming the communication delay on a path $\mathcal{P} = \{(v_0, v_1), (v_1, v_2), (v_2, v_3), ..., (v_{l-1}, v_l)\}$, $\mathcal{P} \subseteq \mathcal{V}$, of length l between two hosts v_0 and v_l, is denoted as $t(\mathcal{P})$. This *path delay* $t(\mathcal{P})$ is defined between two hosts (EPs or DTs). $t(P)$ depends on the hop count l, the transmission behavior $t(u, v)$ on each individual hop $(u, v) \in \mathcal{P}$, and also on the queueing delays at the individual nodes, denoted by the (probabilistic) function $t_{\text{queue}}(u)$. The following equation holds:

$$t(\mathcal{P}) = \sum_{(u,v)\in\mathcal{P}} \left(t(u, v) + t_{\text{queue}}(u)\right).$$

The queueing occurs only at transmitting nodes, thus the destination $v_{n_{\mathcal{V}}}$ does not add any queuing delay to the path. This means, for the sake of simplicity, we deliberately neglect processing delay on the destination node, and consider queueing delay to occur only on transmission queues. The single-hop delay $t(u, v)$ between nodes u and v is a function that is derived from a set of protocol-related values as follows:

$$t(u, v) = (t_{\text{cont}}(u) + t_{\text{arb}}(u) + t_{\text{trans}} + t_{\text{prop}}(u, v) + t_{\text{ack}}) \times \mu(u, v)$$

The values $t_{\text{cont}}(u)$, $t_{\text{arb}}(u)$, t_{trans}, $t_{\text{prop}}(u, v)$, t_{ack}, $\mu(u, v)$ are the contention delay, arbitration delay, transmission delay, propagation delay, acknowledgment delay and the number of transmission attempts at node u, respectively, as we have already introduced in Section 2.3.1.

We define the busy time of node u as the portion of $t(u, v)$ that it is blocking the medium for its transmission. This is the time that is consumed by all protocol-related transmission activities, except from the contention with other nodes.

$$t_{\text{busy}}(u, v) = (t_{\text{arb}}(u) + t_{\text{trans}} + t_{\text{prop}}(u, v) + t_{\text{ack}}) \times \mu(u, v) = t(u, v) - t_{\text{cont}}(u) \times \mu(u, v)$$

The busy time is the portion of which the node u contributes to the busy state of the channel.

The propagation delay $t_{\text{prop}}(u, v)$ is a known quantity and computes as the fraction of the distance $d(u, v)$ between nodes u, v and the speed of light c_M in the given medium.

$$t_{\text{prop}}(u, v) = \frac{d(u, v)}{c_M}$$

Similarly, the transmission time t_{trans} is defined by the Physical Layer protocol and computes as

$$t_{\text{trans}} = \frac{s + h}{\text{tp}},$$

where tp is the throughput on the Physical Layer, s is the payload size of a packet, and h is the header size, involving all layers from Physical to Application Layer. Since we can assume from our flow model that all applications have equally small packet sizes, and the throughput is comparably big, we can assume the transmission delay to be constant. The acknowledgment delay t_{ack} is a constant, too, given the specific Layer 1 and 2 protocols.

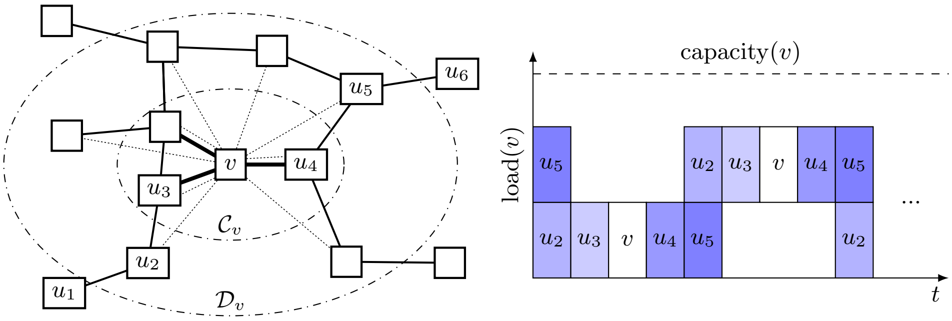


Figure 6.8: Behavior of the channel utilization in a network with a single active flow on the path $\mathcal{P} = (u_1, u_2, u_3, v, u_4, u_5, u_6)$. Highlighted in the diagram is the channel utilization as seen by v, which includes all nodes within $\mathcal{D}_v$. Nodes u_1 and u_6 are not visible to v, so they do not contribute to the utilization at v.

This leaves us with four parameters in the chain of single-hop transmissions that we have to analyze with specific means, namely the contention delays $t_{\text{cont}}(u)$, arbitration delays $t_{\text{arb}}(u)$, and the number of transmission attempts $\mu(u, v)$. All four are heavily dependent on the concrete communication technology i.e., the Layer 1 and 2 protocols involved. Some technologies, for example, do not have a prioritization mechanism that is based on waiting times, so their contention delay may be zero. Some technologies with a TDMA scheme, like 5G, can have a constant contention delay. Similarly, some technologies have different approaches to a reliable data transfer than retransmissions, so $\mu(u, v)$ may be just the constant 1. It is also possible to trade reliability for latency by adjusting the parameter $\mu(u, v)$.

6.4.1 Conflicting Transmissions, Scheduling, and Channel Capacity

With wired communication, there is no reason to consider side effects between the transmission processes on the individual links, and the transmission on a given link does not affect another transmission process on other links. For wireless communication, however, this is not the case, and side effects exist. These are expressed by the conflict graph of the network. When a node v transmits at a given time slot t, then all nodes within detection range $\mathcal{D}_v$ of node v cannot receive packet transmissions from other nodes than v. The channel is said to be blocked by v in that case. The blocking is, however, spatially restricted to the detection range, and nodes outside of $\mathcal{D}_v$ are not affected by v's transmission. As a consequence, the channel has to be arbitrated, and the nodes within $\mathcal{D}_v$ need to delay their transmissions until the channel is free again.

With some wireless technologies, multiple parallel transmissions are possible, too, up to a given *channel capacity* capacity(v). This can be achieved by various means, like multi-channel radio, MIMO, Non-orthogonal Multiple Access (NOMA), or Frequency Division Multiple Access (CDMA). The channel capacity depends, in general, on various parameters, such as frequencies, transmission rate, local interference, noise, humidity, or bandwidth. It is, therefore, necessary to assume that the channel capacity is a local phenomenon and may be different for each node v. An active transmission within the detection range $\mathcal{D}_v$ contributes to the *channel utilization*, which defines at which portion the channel capacity is used at a given time. The utilization cannot exceed 100 %.

The side effects have a negative impact on the series of transmissions that are necessary to transmit packets on a path through the network. Figure 3.3 shows this behavior with an example where an isochronous haptic flow needs to be transmitted on a 6 hop path. The utilization diagram shows the channel as it is perceived by the centermost node v over time t. Although the active flow is admitted with a given network load l, that is given by means of its packet size and packet rate, the actual load that is measured by v is bigger than l, since the flow splits into its individual single-hop components. v, in this example, has four of the participating nodes in its detection range (u_2 to u_5), and has to send packets itself to its next-hop neighbor u_4. This means the network load at v appears to be five times bigger than the allocated load l of the flow.

To represent this phenomenon, we have to introduce a measurement for the local load at v. We thus define the function $\text{load}(\cdot) : \mathcal{V} \to \mathbb{N}$. The relations $\text{load}(v) \geq 0$ and $\text{load}(v) \leq \text{capacity}(v)$ must hereby hold. Both relations are enforced using the Layer 2 protocol, which arbitrates the channel between the nodes.

Although the relative load must meet the channel capacity locally at each node, the same is not true for mutually distant transmissions that are outside of each other's range and no side effects occur. This means, that distant nodes can transmit simultaneously, even when they are part of the same WMHN. This phenomenon is called *spatial reuse* and is a design feature of various Layer 2 protocols, such as 5G and IEEE 802.11.

6.4.2 Latency Prediction by Probabilistic Modeling

The delay $t(\mathcal{P})$, as well as most of its components, is a function of many complex interactions on all communication layers. Only for a few, if any, protocols these values are deterministic. They vary over time, with each packet, and between each node and edge, often in a non-deterministic manner. Often, protocols themselves include non-deterministic aspects, as CSMA MAC schemes do, for example. A prediction can, therefore, only be made probabilistically. For prediction models, we have to transform the non-trivial components of the delay into random variables as follows. The variable T_P denotes the path delay, Q_v the queueing delay, $T_{u;v}$ the single-hop delay, A_v the arbitration delay, B_v the contention delay, and $M_{u;v}$ the number of transmission attempts undertaken. The variables have the same relation as the original functions, i.e.,

$$\begin{aligned} T_{\mathcal{P}} &= \sum_{(v,u)\in P} Q_v + T_{u;v}, \\ T_{u;v} &= (B_u + A_u + t_{\text{trans}} + t_{\text{prop}}(u,v) + t_{\text{ack}}) \times M_{u;v}. \end{aligned}$$

Furthermore, we define a probabilistic version of the analytical busy time $t_{\text{busy}}(u,v)$, $D_{u;v}$, that excludes the contention delay:

$$D_{u;v} = (A_u + t_{\text{trans}} + t_{\text{prop}}(u,v) + t_{\text{ack}}) \times M_{u;v} = T_{u;v} - B_u \times M_{u;v}$$

The random variables are, in general, drawn from individual, but not independent, general random distributions. Depending on the underlying protocols, they can be continuous random variables (e.g., with the unslotted ALOHA MAC scheme), or, as is mostly the case, discrete random variables, given a fixed slot time t_{slot}. The prediction is now concerned to infer as much information as possible about the random distributions, for example, the type of distribution, the distribution parameters, or indicators as the expected value, the variance, and the modes. A full representation is also given if the probability density function (PDF) is known for continuous, or the probability mass function (PMF) for discrete variables, or the cumulative distribution function (CDF). These functions can be either obtained analytically with protocol analysis, by a probabilistic model (e.g., Bayesian inference), or numerically by sampling.

Assuming that we can provide the probability density functions (PDFs) for all components of $T_{\mathcal{P}}$, the above model provides a rule to derive the PDF of $T_{\mathcal{P}}$, too. The model invokes only summation and multiplication operations. The PDF of a sum of two independent random variables $A + B$ can be obtained by the convolution of the PDFs of A and B [94]:

$$f_{A+B}\left(c\right) = \int_{-\infty}^{\infty} f_A\left(c\right) f_B\left(b - c\right) \mathrm{d}b = (f_A \circledast f_B)(c).$$

Similarly, a solution for $A \times B$ exists for two independent random variables A and B:

$$f_{AB}(c) = \int_{-\infty}^{\infty} f_A(c) f_B\left(\frac{b}{c}\right) \mathrm{d}b.$$

The derivation of the multiplication rule is analogous to the derivation of the summation rule.

Both the addition and the multiplication rule require statistical independence and do not hold for dependent random variables $(f_{A;B}(a;b) \neq f_A(a) f_B(b))$. We have to show that independence can be assumed, at least by approximation to a certain degree.

6.4.3 Dependencies Between the Random Variables

We will axiomatically assume the case that there is only one class of dependency between the variables. First, for each node $u \in \mathcal{V}$, the contention delay B_u depends on the busy times $D_{u;v_i}$ of all nodes $u_i \in \mathcal{V}$ within the network. Second, the queueing delay Q_u depends on all busy times $D_{u;v_i}$, as well as on the contention delays B_{v_i} of all nodes $u_i \in \mathcal{V}$. Thus, unknown functions f_u exist that describe the contention and queuing delays as

$$B_u = g(D_{u;v_1}, ..., D_{u;v_{n_\mathcal{V}}}), \text{ for } v_i \in \mathcal{V},$$

$$Q_u = f(D_{u;v_1}, ..., D_{u;v_{n_\mathcal{V}}}, B_{v_1}, ..., B_{v_{n_\mathcal{V}}}), \text{ for } v_i \in \mathcal{V}.$$

In this model, queueing delay results from the circumstance that the medium is shared among nodes. Its service time associated with the queueing delay is a random variable, the busy time $D_{u;v}$. At a rate of $\frac{1}{D_{u;v}}$, the medium can process one packet. This assumption is different from most of the approaches that can be found in current literature, where the queueing delay is often modeled as an independent entity that is determined merely by the processing power of the network routers (e.g., [95, 96]). However, we find this independence assumption insufficient for WMHN. Wireless networks exhibit increasing node densities, while at the same time, transmission ranges remain constant or increase as well. Topology and conflict graphs are thus often dense, too, and the amount of nodes that are blocked during a single transmission phase grows significant. The bottleneck of WMHN is, therefore, often not the processing speed of its router nodes, but the medium utilization. The queueing delays result from the many accesses to a restricted shared resource that we have to investigate in more detail in later sections, finally, model the functions g_u and f_u in Chapter 8.

Using the distinction between busy time $D_{u;v}$ and the single-hop transmission time $T_{u;v}$, we can assume that the dependency of all other random variables is merely a subject of the internals of the used protocol. For example, we assume that arbitration delays A_u, required number of transmission attempts $M_{u;v}$, and the individual queueing delay Q_u on different nodes are independent for most protocols as WiFi or cellular networks. Therefore, we axiomatically assume the following independence for arbitrary nodes $u, v, w, x \in \mathcal{V}$:

- A_u and A_v are statistically independent,
- $M_{u;v}$ and $M_{w;x}$ are statistically independent,
- A_u and $M_{v;w}$ are statistically independent, and
- $D_{u;v}$ and $D_{w;x}$ are statistically independent.

The resulting process model for a probabilistic delay analysis is sketched in Figure 6.9. Within the process, estimates $\hat{b}_v, \hat{m}_{v;u}, \hat{a}_v, \hat{t}_{v;u}, \hat{q}_v$ of some form have to be found for the respective random variables $B_v, M_{v;u}, A_v, T_{v;u}, Q_v$, finally to obtain an estimate $\hat{t}_v$ for T_v. The estimates can be of various forms and do not have to be scalar values – including sampling of the PDFs of the random variables. The most complex part of the estimation process is the determination of the queueing delay due to its unknown dependency relation to all the other

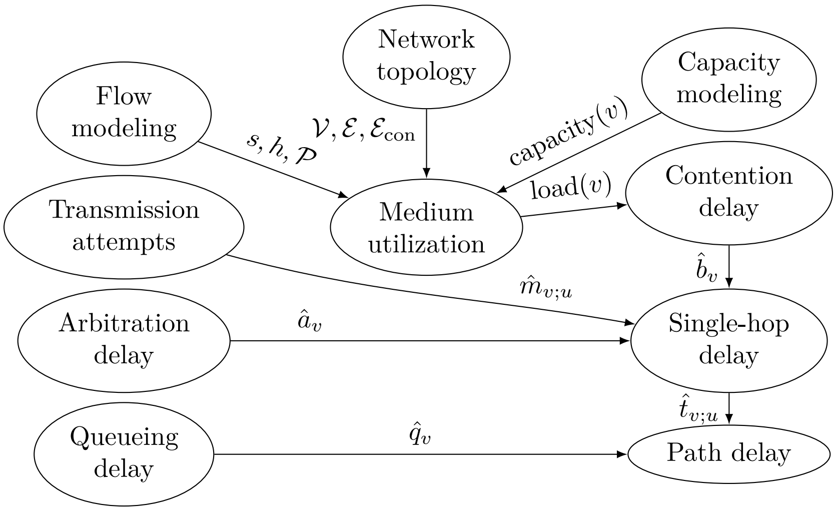


Figure 6.9: Process for obtaining an estimate $\hat{t}_{\mathcal{P}}$ of the end-to-end path delay $T_{\mathcal{P}}$.

variables. For each of the other random variables, the distribution can be modeled directly from the used protocol.

The queueing delays have to be modeled by a queuing model that relates the medium utilization at a given node to its network load. A trivial relation is often not applicable here, so we will develop a model derived from queueing theory and a Markov model that considers the interdependence of individual sending nodes on their shared medium.

6.5 Queueing Model

The axioms that we have introduced above introduce the necessity of a queueing model that explains the shared medium access as a random process. The queueing delay Q_v on a node v can be then modeled using queueing networks, which are a common analysis tool for computer networks [97, 98].

Similar to wired networks, WMHN can be viewed as a system of queues, where each node has access to a shared medium. Having identified a queueing model, we can determine the queueing delay Q_v as the waiting time within the queue of node v. It can be determined either in terms of its expected value by Little's Law [99], or, by the specific probability distribution of the waiting time of the queueing system. For an $M/M/1$ queue, for example, the latter is known to be a truncated exponential with the cumulative distribution function (CDF)

$$F_{Q_v}(t) = 1 - \rho e^{-t\mu(1-\rho)}.$$

Here, $\rho = \lambda/\mu$ is the queue utilization, λ is the arrival rate of jobs in unit time, μ is the service rate of the queue. The mean of this distribution is

$$\mathrm{E}[Q_v] = \frac{\rho}{\mu - \lambda}.$$

Unlike wired networks, the queueing model for WMHN does not only depend on the processing power of the node. The bottleneck here is the shared medium itself, and the constraining resource is the medium time. Assume a node v wants to send a packet to a neighboring node u. The medium access by v requires that all nodes in the collision range $\mathcal{D}_v$ need to be prevented from medium access during the transmission. This is because all the nodes in $\mathcal{D}_v$ can interfere with that transmission at u's receiving antenna, and thus they need to be silent during v's transmission. The service process, therefore, is the MAC scheme that is responsible for scheduling the transmissions on the medium. The following Figure 6.10 shows this mechanic with a transmission $v \to u$. Thus the channel situation within the region $\mathcal{D}_v$ translates to a queueing model that is shown on the right hand side of Figure 6.10.

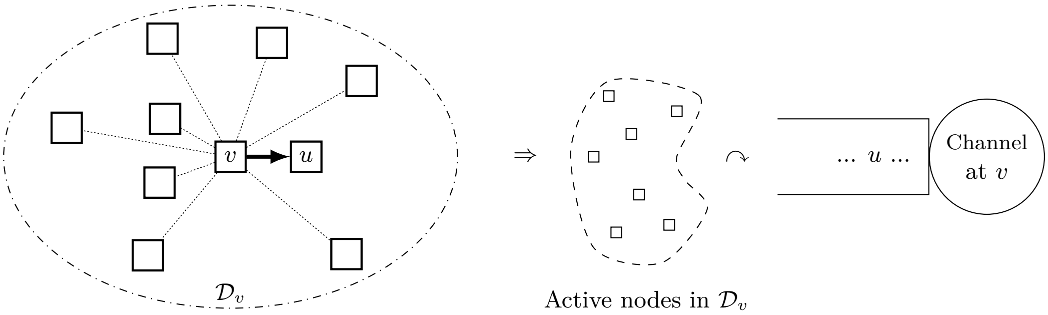


Figure 6.10: Relation of the channel utilization to the conflict graph. Node v has a transmission to node u scheduled. v queues the transmission according to the state of the channel that it perceives locally.

The sending node v now needs to be scheduled to access the medium and thus enters a queueing system. In the queue, all nodes that want to access the channel to send a packet (with any destination) have to be coordinated. As not every node might be active at a given time, u might encounter a queueing system with smaller utilization, or higher utilization, depending on the current situation.

For all nodes in $\mathcal{V}$, there exists a similar, but individual queue for the channel access, since the medium appears differently for each node. Necessarily, though, the queue's service processes have to be coordinated, which means that the queueing model cannot be as simple as a Poisson process of a *M / M / 1* queue. In theory, a specific queueing system can be affected by any other scheduled transmission within the same WMHN, and even transmissions outside of $\mathcal{D}_v$ might affect the servicing at node v. A theoretic boundary does not exist in general, for example, CSMA-based networks can suffer from starvation problems [100], and nodes may not have the chance to send packets at all in certain situations. This means, that in our queueing model there exist $n_{\mathcal{V}}$ such queues, one for each node in a WMHN of size $n_{\mathcal{V}}$.

To solve this complex problem, we derive a queueing model based on an existing throughput model first introduced by Stojanova et al. [100], that we design in Chapter 8. We leverage the fact that the queueing delay is related to the utilization ρ (here, the *channel utilization*) in the way described above. This way, we avoid modeling each packet's path throughout the network, and can focus on modeling the behavior of the individual nodes.

PART III

Access Networks in the Tactile Internet

CHAPTER 7

Modeling Latency of IEEE 802.11 Access Networks

In this chapter, we create a probabilistic lantency model IEEE 802.11 wireless multi-hop networks, based on our formalization. WiFi networks are widely used in Smart Homes, as enterprise networks, and for public access. At the same time, the complexity of the channel access scheme makes WiFi networks a good benchmark for a modeling approach. The dominant market role of WiFi IEEE 802.11 can be addressed directly to the adoption of a random access scheme for the MAC layer, as the flexibility and low configuration overhead is essential for the market role that WiFi operates in. WiFi is the most widely used representative of Wireless Local Area Networks (WLANs).

Even as WiFi networks are now starting to be replaced by 5G networks, the value of a WiFi latency model is still high. For example, 5G allows the inclusion of ISM-band frequencies where very similar channel access schemes will be adopted. Since 5G also allows for multi-hop communication, its behavior is likely very similar.

We start by surveying the related work on WiFi latency models, and then review the basic functioning of the WiFi channel access scheme. Secondly, we contribute a latency model for WiFi WMHN that allow for assessing of the latency distribution in small-scale WiFi networks. The model, at first, ignores the queueing delays on the nodes, as we outline a separate queueing model in the next chapter. However, the model already shows capable of modeling the latency in small-scale WiFi networks, as the simulation analysis presented in this chapter suggests.

7.1 Existing Methods and Related Work

Modeling techniques for computer networks were thought of and have been used virtually since the beginning of modern telecommunication. Modeling tools, like queueing analysis, Markov models, probabilistic modeling, statistical analysis, Petri nets, and others, are described in various textbooks [14,64,97,98]. However, despite all these techniques, the modeling of computer networks is, and always has been, a research task that involves much experience, proper modeling, and excessive parameter calibration. The tools are more general approaches for modeling, and they simplify the underlying processes to the degree that allows for quick evaluation.

Computer networks (as well as computer systems in general), can be evaluated using three different techniques; mathematical analysis, measurement, and simulation [64]. Measurement requires that a system has already been realized, and it can be equipped with sensors while operating with the intended workload in order to determine the value of interest. With mathematical analysis, in contrast, the system may not yet exist but is modeled with mathematical tools, like Markov chains, queueing models, or Petri nets. These can then be analyzed for the required properties. Mathematical models must often abstract the real world in order to be of manageable complexity, and thus are often simplifying by omitting details or assuming facts and causal relationships. However, they are good to assess turning points and corner cases in systems. For analyzing worst cases in terms of some goal function, a mathematical analysis is often the only viable technique.

A third technique is available with simulation. Here, a system is approximated by theoretical constructs, similar to those used for mathematical analysis. But in contrast to mathematical analysis, the model is numerically sampled, by some random number generation for all workload and decision processes within the model. A simulation always involves the calculation of numerical output by processing model internal relations, which can be either discrete or continuous in time.

In terms of probabilistic models, a mathematical analysis would be concerned with the finding of some distribution parameters, or with the determination of PDF of a given random variable related to the system. A simulation, in contrast, would be concerned with sampling the random variable, providing sample means, variances, and a histogram.

Modeling techniques for ANs involve the modeling of WMHN in different ways. However, the modeling of WMHN in terms of accurate mathematical analysis is an open problem in literature until today. Although WMHN are quite well understood in terms of throughput, their behavior in terms of latency is only vaguely understood. This is mostly because of the complex interplay between nodes on the Data Link Layer, which we have already laid out in the formalization. Latency analysis is more complex, though, due to the queueing behavior that has to be considered (which is not relevant for throughput analysis). But the throughput models are interesting here, too, as they show a variety of modeling approaches that can be interesting for latency analysis, too.

7.1.1 Challenges Specific for the Tactile Internet

Our following literature review reveals three major challenges for WMHN performance modeling. Most of the modeling attempts in literature address one or two of these, but to the best of our knowledge, no approach covers all three.

Multi-hop networking. Many analyses, especially early work, cover mostly single-hop communication. Most papers concentrate on infrastructure networks, where an AP communicates with one or more client nodes. This has been the main operation mode of both IEEE 802.11 and telecommunication networks to the time and therefore was a sufficient approach in the past. Multi-hop communication, which is gaining more importance today, is more difficult to model, as channel coordination is more complex, and nodes tend to affect each other far beyond their communication ranges. Models that cover multi-hop communication thus have to cover the whole network at once, where the complexity increases at least quadratically with the number of nodes.

Modeling queueing delays on the nodes is related to the multi-hop problem, as the end-to-end delay must include the waiting times that the individual packets wait in buffers within the routers. It is, therefore, not sufficient to model the individual single-hop-delays on their own. This adds even more complexity compared to the multi-hop challenge since individual packets must be tracked along their path through the network.

Non-saturation, finally, covers the aspect that a network may not be fully saturated, and the model has to respect the current network load. For early throughput analysis, it was sufficient to model the case where all network nodes are fully saturated, i.e., all nodes have at least one packet in their transmission queue at any given time. For delay models, however, the network saturation is a parameter of the end-to-end delay, as we already have elaborated. Modeling non-saturated networks adds another layer of complexity in order to deal with the different load states of the nodes, and proper workload models have to be incorporated.

7.1.2 Throughput Modeling for IEEE 802.11 Multi-Hop Networks

Throughput analysis for IEEE 802.11 networks have been a subject of interest since the early adoption phase, and more complex scenarios have been covered in the literature. They include multi-hop models, as well as non-saturated networks. The throughput of WiFi networks was the central problem for early WiFi installations, as the data rate was the single key performance indicator for the dominant multimedia content up to the late 2010's.

In 2000, Bianchi [101] pointed out that the throughput of IEEE 802.11 infrastructure networks is unstable, and increasing network loads beyond a certain threshold can result in a decline of the cumulated throughput in a network. In their seminal paper, they presented an initial Markov model for the binary exponential backoff procedure, which resembles the core aspect of CSMA/CA medium access for IEEE 802.11. Each state in the model is a two-dimensional vector, where the first dimension represents the backoff counter, and the second the backoff stage. Their model can predict the asymptotic cumulated throughput of a saturated infrastructure network, i.e., all stations are considered to be sending with a constant load of 100 %. The model was evaluated by simulation for some large-scale networks, although no multi-hop characteristics have been evaluated. As an early work to represent the behavior of large-scale IEEE 802.11 networks, its main contribution is the modeling approach, which was adopted by many of the following research papers.

Frohn et al. [102] provided another study investigating the effective throughput of 802.11 n networks using discrete Markov chains. Their model also includes two different channel error models, and also considers the aggregation of long frames and block acknowledgments, which is an integral function of IEEE 802.11 in infrastructure networks. The different payloads of frames result in a certain modeling complexity.

Wu et al. [103] presented a throughput optimization using a similar modeling approach with Markov chains. Similar to Bianchi, they found that the achieved throughput strongly depends on the IEEE 802.11 MAC parameters settings, especially on the contention window size, and proposed a method for optimizing the parameter settings.

Cali et al. [104] have developed an algorithm that can adapt the contention window size at run-time to optimize network throughput. They have shown that realized network throughput can be far from the theoretical maximum.

Chatzimisios et al. [105,106] have verified Bianchi's approach with extended modeling using the OPNET simulation tool. They provide a detailed analysis of the relation between the contention window size and the achieved throughput in a simulated IEEE 802.11 network.

Maadani and Motamedi [107] presented a model that extends that of Bianchi by a third dimension, the inter-frame space waiting phase. With this, they cover a specific issue with backoff freezing that causes throughput fluctuations. They also extend Bianchi's analysis with a delay model., however, the model still assumes full saturation.

Finally, Stojanova et al. [100] proposed a conflict graph based Divide and Conquer model for 802.11 infrastructure networks. They also use discrete Markov chains, but model individual activation states of nodes, so that they do not assume a fully saturated network. It considers the so-called *regime* of a node, which is a binary state that signals if a node has a packet in its transmission queue, or if it has no packet to transmit and thus is idle at a given time instant. The model divides the network into subnetworks, where each subnetwork covers a specific permutation of the node regimes, thus solving the throughput problem for each activation state. The overall network throughput is then calculated by a linear combination of all the subnetworks, given their occurrence probabilities. Similar to the approach of Zocca et al., it is not capable of modeling multi-hop paths through a network, as only APs and their according client stations are considered. It also does not consider retransmissions.

The Divide and Conquer approach from Stojanova et al. provides an elegant way to cope with the non-saturation assumption by calculating a solution for each activation state. Their approach, therefore, provides an abstraction from the saturation problem, and the individual subnetworks can be solved under the saturation assumption, which allows for a tremendous simplification of analysis. We cover the approach of Stojanova et al. in detail in Chapter 8, where we use it as a basis to solve the modeling problem for queueing delay in multi-hop networks.

7.1.3 Latency Modeling for IEEE 802.11 Multi-Hop Networks

Latency models are the subject of more recent literature than throughput analysis. Some of these publications are based on Bianchi's approach, but the delay modeling is then either restricted to single-hop delays or fully saturated networks.

Sakurai and Vu [108] have presented a stochastic model for the single-hop delay $T_{u;v}$ in a fully saturated IEEE 802.11 network with $n_{\mathcal{V}}$ nodes and perfect channel conditions. It can be used to sample the CDF of the contention delay numerically. The mean, variance, and asymptotic behavior can be analyzed by their model as well. It was evaluated with a simulation.

Vardakas et al. [109], as well as Raptis et al. [110] both use Bianchi's Markov modeling technique to infer the MAC delay distribution. Vardakas et al. also analyze the mean queueing delay and the mean end-to-end delay under the assumption of a *M / G / 1* system. However, both approaches assume full saturation.

Tian and Tian [111] have analyzed the performance of IEEE 802.11 Distributed Coordination Function (DCF) for soft real-time applications with a periodic traffic pattern. Their basis is also Bianchi's approach which they extend by introducing idle states for non-saturated traffic and apply a timing model to obtain the delay distribution.

Zocca et al. [112] found that the behavior of mean delay in grid networks of size $L \times L$ with load ρ is $\mathcal{O}((\frac{1}{1-\rho})^L)$. The analysis is based on the hard-core model of statistical physics.

A multi-hop analysis was presented by Kanematsu et al. [113] in 2020, where they have developed a Markov model for analyzing WMHN delay and throughput for a linear network. A linear network consists of only one source and one destination node, with several relay nodes in between. The relay nodes are arranged linearly, such that each node has exactly two neighbors and statically forwards packets in just one direction. Despite the simplified topology, they differentiate between the communication range ($\mathcal{C}_v$) of a node and its detection range ($\mathcal{D}_v$). They were able to model the end-to-end delay by a Divide and Conquer approach where they used distinct Markov models for each node to calculate the transmission times, collision probabilities, and queueing delays. They also assume a certain arrival process at the sender, which means they do not assume a fully saturated network.

A latency model that is capable of modeling an arbitrary IEEE 802.11 WMHN, with an arbitrary topology and more than one existing data flow is a subject that has yet to be investigated. To the best of our knowledge, no modeling approach in such a direction has been presented in literature yet.

7.1.4 Modeling Wireless Networks as Poisson Processes

We want to mention another method that has been used for analyzing computer networks for a long time, which is the modeling of Poisson processes. A Poisson process is characterized as a queueing system where both the arrivals and the departures are exponentially distributed. In Kendall's notation, these are designated as *M / M / 1* or *M / M / N* (for multi-server systems). These models fit very well to wired computer networks, where the shared nature of the wireless medium does not take effect. But they are also applicable for wireless systems, especially when point-to-point communication links can be assumed.

The idea is to model each router in a multi-hop communication system with its own Poisson process, that can be concatenated when multi-hop connections are to be analyzed. The input rates come from the applications, while the output rates are defined by the communication links. The utility of Poisson processes lies in their simple modeling and the well-developed and simple analysis. Concatenations of Poisson processes of the same output rates are known to obey the Erlang distribution, and, more generally, processes with different output rates follow the Gamma distribution. Both Erlang and Gamma distributions are well understood, although the Gamma distribution requires numerical analysis due to the lack of a closed formula. A huge benefit of these models is that the queuing delay is modeled directly. The end-to-end delay thus results directly from the model.

Poisson modeling is very effective with telecommunication systems, where multi-hop connections on the backhaul network are often characterized by point-to-point communication links. Jaber et al. [95] proposed such a model for modeling wireless delay, throughput, and resilience for 5G wireless backhaul networks. The approach models delays using probability distributions for retransmission and processing delays, assuming ideal channel scheduling. The underlying network model assumes a star topology with point-to-point communication links between a gateway station and multiple base stations, which are interconnected by different routers.

Zhang et al. [96] use a similar model for 5G backhaul modeling. They investigate different communication frequencies, where for the sub-6-GHz frequency range they use the interference to model the circumstance that routers can affect each other's transmissions. They model the reception probability by considering the link budget between the nodes, including interference from other nodes. While this interference model keeps the simplicity of the modeling approach, it does not consider the circumstance that the medium time is shared between nodes that are in mutual communication range.

The Poisson modeling techniques are appropriate for telecommunication systems that have big enough capacities in terms of channel resources. They assume the bottleneck of the network to be the processing power of the individual nodes instead of the limited channel resources. When channel resources are limited, especially when all nodes have to use the same communication channel the medium time becomes the main bottleneck, and the queueing occurs mainly due to the complex processes of the medium.

7.2 Medium Access in IEEE 802.11 Networks

IEEE 802.11 was first introduced in 1997 as a wireless counterpart to Ethernet Local Area Networks (LANs). It has evolved in many iterations and has received various amendments since then. The standard is regularly updated to a revision that incorporates the latest amendments. Currently, the latest revision is known as IEEE 802.11-2020 [39], which incorporated all amendments up to IEEE 802.11aq. The standard defines both MAC sublayer and Physical Layer, leaving the Logical Link Layer (LLC) (a sub-layer of the Data Link Layer) subject to other 802.*xx* standards. Different Physical Layers have been defined (some of them later revoked) for various ISM bands, e.g., on 2.4 GHz, 5 GHz, and 60 GHz bands.

The classical architecture of IEEE 802.11 is the *infrastructure network* that consists of a set of APs, that serve a varying number of client nodes (called stations in the IEEE 802.11 terminology) within their range. Each AP forms a single-hop network referenced as a Basic Service Set (BSS), which is the identifier of the network. Various BSS can coexist on the same channel at the same location. APs can be connected to form a larger network using a *distribution system*, which can be wired or wireless. When more than one AP join in such a form, the network is identified by an Extended Service Set (ESS). Besides the infrastructure mode, als an *ad hoc* mode and a *mesh* mode are available. The ad hoc mode can be formed between stations without the presence of an AP. The mesh mode finally introduces real multi-hop capabilities, which we will have to investigate further during this section.

The Physical Layer offers the service of a Clear Channel Assessment (CCA), which senses the channel prior to sending and reports if the channel is sensed idle (free) or busy (occupied by another carrier). The CCA is a necessary mechanism for the random access property, as it is used to defer a transmission in case of a busy medium. In case of a free channel, a waiting transmission is always carried out, no matter what state the neighboring stations are in – no reservation or any other means of scheduling happens. This ensures that, even in a network with many idle nodes, the overhead in waiting time due to the network size is low.

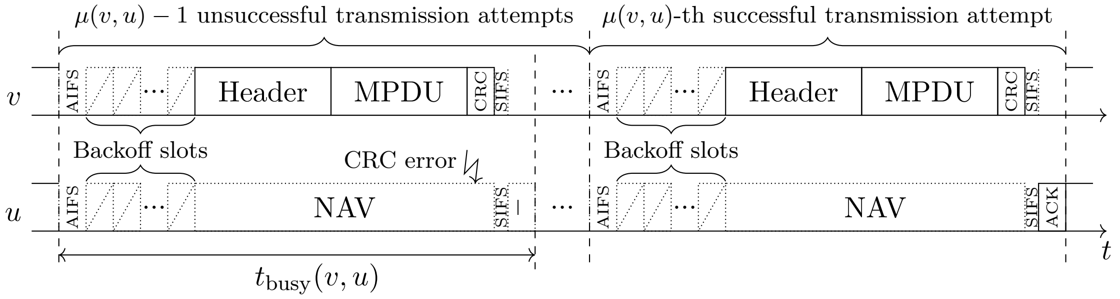

$\mu(v,u)-1$ unsuccessful transmission attempts
$\mu(v,u)$-th successful transmission attempt
v
AIFS
Header
MPDU
CRC
SIFS
Backoff slots
CRC error
u
NAV
ACK
$t_{\text{busy}}(v,u)$
t

Figure 7.1: Timing diagram of an IEEE 802.11 EDCA transmission process between two nodes v and u. Solid boxes indicate active sending states, while dashed boxes indicate waiting times for either backoff slots, Inter-Frame Spaces (IFSs) or Network Allocation Vectors (NAVs).

IEEE 802.11 is a slotted protocol, where a *slot time* t_{slot} is typically just the length of several symbol durations. If the medium is sensed busy by the CCA for one slot time, an exponential backoff mechanism takes effect to resolve for the then necessary step of channel arbitration. Nodes wait for a random number of slot times between 0 and a variable called the *contention window size*. With each collision that occurs, the affected nodes double the contention window size, which sets down the priority of their next attempt in favor of other nodes. The basic principle is illustrated in Figure 7.1. The number of slots that a node defers its transmission is called *backoff slots*. The payload of the MAC layer, the MAC Protocol Data Unit (MPDU), is prepended with its header information and extended with a CRC checksum. When a node v 'wins' a contention phase by drawing the shorter number of backoff slots to wait, it begins sending, and in the following slot, competing nodes sense the transmission by means of CCA. They also further defer their next transmission by calculating the amount of time that v will block the channel (stored within a register called Network Allocation Vector (NAV)), and possibly adapt their contention window size.

There have been many variations of this mechanism in the standard, from which all of them are still supported in terms of backwards compatibility. The current, most adopted mechanism is called Enhanced Distributed Channel Access (EDCA), which was introduced with IEEE 802.11q in 2005. It is an extension of the original DCF specified in the early 1997 standard. EDCA has itself received adaptations to support mesh networks. Additionally, we have ourselves proposed an extension for Haptic Communication with the Tactile Coordination Function (TCF) [114], which we further describe in Chapter 9.

All these channel access functions rely on defined waiting times of different sizes, called Inter-Frame Spaces (IFSs), to coordinate for channel access. The shortest IFS, the Short Inter-Frame Space (SIFS), is the reserved waiting time after which a receiving nope sends an acknowledgment to the sender. Its short waiting time ensures that the acknowledgment is not interrupted by another node with any more urgent transmission. Regular transmission must wait at least the (slightly longer) Arbitration Inter-Frame Space (AIFS), before the counting of the backoff slots begins. The distinction between SIFS and AIFS is sufficient for proper ordering of frames in terms of regular operation, while some other IFSs are defined to e.g. send configuration information from the AP.

7.3 Modeling Latncy for WiFi WMHN

Since we have developed and elaborated a general network model for Access Networks, we now further investigate on special network technologies in detail. It is an example model that is applicable only for small-scale networks. However, it is intended to represent all necessary aspects of the Layer-2 mechanics of the WiFi protocol, which proves our probabilistic modeling approach. It omits the effects of queueing, for which we derive a more complex model in the next chapter.

7.3.1 Single-Hop Delay

In IEEE 802.11 networks, the (random) single-hop delay $T_{v;u}$ comprises of random contention delay B_v and arbitration delay A_v, as well as constant transmission delay t_{trans}, propagation delay, and acknowledgment time t_{ack}, respectively, multiplied by the random number of transmission attempts $M_{v;u}$:

$$\begin{aligned} T_{v;u} &= (B_v + A_v + t_{\text{trans}} + t_{\text{ack}}) \times M_{v;u} \\ &= (B_v + D_{v;u}) \times M_{v;u}. \end{aligned}$$

As elaborated in Chapter 6, the busy time $D_{v;u}$ summarizes all the components that are necessary for one frame transmission and are defined by the standard. We will now describe these components for IEEE 802.11, which we already have sketched in Figure 7.1 earlier in this chapter. We neglect the propagation delay at first in this model since in WiFi it is specified to be less than a slot time. In other words, signals reach their destination always within the same slot.

Arbitration in slotted CSMA/CA depends on the drawn number of backoff slots α_v, which is drawn from a uniform distribution, plus the duration of the AIFS which is a means for prioritizing the respective Access Category [114].

$$A_v = t_{\text{AIFS}} + \alpha_v \times t_{\text{slot}}$$

The transmission time computes from to the MPDU size s plus the header length h as

$$t_{\text{trans}} = \frac{s}{r_v} + \frac{h}{r_{\text{base}}}.$$

The contention delay is the amount of time that a node waits on a busy medium until it can transmit. For inferring this time we consider the number of contending nodes in the neighborhood $\mathcal{D}_v$ of v. This estimation will serve as a simplification until we introduce our ME queueing model in the next chapter, which can infer the required time more methodically. Let assume the set $\Delta_v \subset \mathcal{D}_v$ of nodes that win the contention before v, such that v must wait for their finished transmission before taking turns on the medium. The contention time of v then computes as

$$B_v = \sum_{u \in \Delta_v} d(u).$$

With a homogeneous distribution of node locations and of local sending conditions, we can estimate the $d(u)$ to be nearly equal to $t_{\text{busy}}(v,u)$, thus approximating $t_{\text{cont}}(v)$ with

$$B_v \approx \delta_v \times d(v),$$

where δ_v is the number of nodes that outrun node v in the contention phase.

The final computation yields

$$\begin{aligned} T_{v;u} &\approx (\delta_v \times d(v) + d(v)) \times M_{v;u} \\ &= M_{v;u} \times (1 + \delta_v) \times d(v) \\ &= M_{v;u} \times (1 + \delta_v) \\ &\quad \times \left(t_{\text{AIFS}} + t_{\text{ack}} + \alpha_v \times t_{\text{slot}} + \frac{s}{r_v} + \frac{h}{r_{\text{base}}} \right) \end{aligned}$$

The terms $M_{v;u}$, α_v and δ_v are random variables, whose distributions we now examine individually.

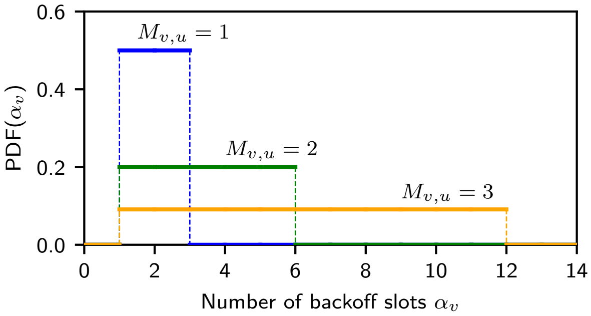


Figure 7.2: Probability density function of α_v with $Cw_{\text{min}} = 3$ at different transmission attempts $M_{v;u}$. The distribution gets broader for later transmission attempts, resulting in longer expected delay.

7.3.2 Backoff Factor

The number of backoff slots α_v is a discrete random variable which is defined in the standard as a uniformly distributed value within the bounds $[1, Cw(m)]$. In the exponential backoff mechanism of 802.11, the value $Cw(m)$ starts at an initial value of Cw_{min}. It is doubled with each unsuccessful transmission, until a fixed maximum Cw_{max} is reached. Thus, the current contention window size $Cw(m)$ is defined by $Cw(m) = \min(Cw_{\text{min}} \times 2^{m-1}, Cw_{\text{max}})$. The number of backoff slots α_v and the number of transmission attempts $\mu(v, u)$ are therefore correlated:

$$\Pr\{\alpha_v = a | M_{v;u} = m\} = \begin{cases} \frac{1}{Cw(m)-1} & \text{for } 1 \leq a \leq Cw(m) \\ 0 & \text{otherwise} \end{cases}$$

Figure 7.2 shows a plot of the distribution for different transmission attempts. The latency increases with each additional transmission attempt as the backoff window gets wider. The joint probability for $\mu(v, u)$ and α_v are according to Bayes' theorem:

$$\begin{aligned} &\Pr\{\alpha_v = a, M_{v;u} = m\} \\ &\quad = \Pr\{M_{v;u} = m\} \times \Pr\{\alpha_v = a | \mu(v, u) = m\} \\ &\quad = \begin{cases} \frac{p_{\text{per}}^{m-1}(1-p_{\text{per}})}{Cw(m)-1} & \text{for } 1 \leq a \leq Cw(m) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

7.3.3 Contention

The contention delay depends on the involved number of nodes that compete for transmission. All nodes contending for the medium draw a random number and wait for this specific number of time slots. Given that two nodes will unlikely draw the same number, the chance for a collision is very low. The conflict graph of the network hereby represents the set of nodes that cannot send at the same time and therefore have to contend in this way before accessing the medium. A node v is adjacent to a node u in the conflict graph, iff $u \in \mathcal{D}_v$. Assume the probability that a node v is in the state that it has a frame to send. Then the probability that exactly c nodes out of the set $\mathcal{D}_v$ are waiting to send a frame follows a Binomial distribution:

$$\Pr\{\delta_v = c\} = \binom{|\mathcal{D}_v|}{c} p_{\text{cont}}^c (1 - p_{\text{cont}})^{|\mathcal{D}_v| - c},$$

where $\binom{\cdot}{\cdot}$ represents the binomial coefficient. In Figure 3.3 the communication and carrier-sense ranges are depicted as the sets of nodes $\mathcal{D}_v$ and $\mathcal{C}_v$, respectively. The connectivity set $\mathcal{C}_v$ defines the network topology, while the set $\mathcal{C}_v$ defines the conflict graph of the network

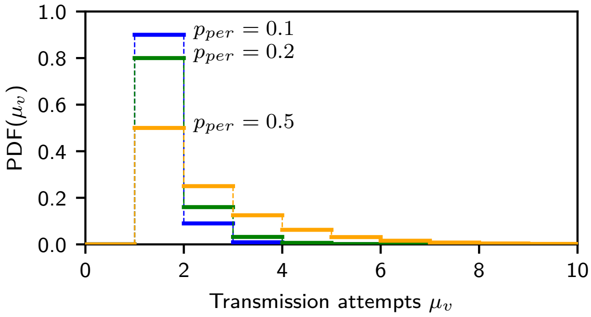


Figure 7.3: Probability density function of $M_{v;u}$ for different packet error rates p_{per}.

which contains a link for every pair of nodes that can potentially interfere with each other. The conflict graph of a IEEE 802.11-based WMHN can be computed online [115].

7.3.4 Retransmissions

The random variable $M_{v;u}$ specifies the number of retransmissions necessary. It is drawn from a discrete random distribution. $M_{v;u}$ can only take natural numbers. It is limited by the maximum number of retransmissions $\hat{m}$. The retransmission count follows a geometric distribution, i.e.

$$\Pr\{M_{v;u} = m\} = p_{\text{per}}^{m-1}(1 - p_{\text{per}}).$$

The respective probability density function of $M_{v;u}$ is plotted in Figure 7.3. As can be seen, even with a relatively high packet error rate of 20 %, the probability density approaches zero quickly, such that the probability that a seventh transmission attempt is needed at all is very low.

The probability $p_{\text{succ}}(\hat{m})$ of successful one-hop transmission of one packet is defined by the number of transmission attempts $\hat{m}$ and packet error rate p_{per}:

$$p_{\text{succ}}(\hat{m}) = \Pr\{M_{v;u} \leq \hat{m}\} = \sum_{r=0}^{\hat{m}-1} p_{\text{per}}^{r}(1 - p_{\text{per}})$$

7.3.5 Probability densities of $T_{v;u}$ and T_P

For calculation of the overall probability density functions we discretize the time steps to be multiples of the IEEE 802.11 Slot Time. Algorithm 1 shows the calculation of the PDF of $T_{v;u}$ including all aforementioned considerations.

Algorithm 1 Calculate PDF of $\Pr\{T_{v;u} = i \times t_{\text{slot}}\}$ for each slot $i \in \mathbb{N}$

```
$pdf_{T_{v;u}} \leftarrow 0, 0, 0, \dots$
for all $m \in [1, \hat{m}]$ do
  for all $a \in [1, Cw(m)]$ do
    for all $c \in [0, |\mathcal{D}_v|]$ do
      $i \leftarrow m \times (1 + c) \times (t_{\text{AIFS}} + t_{\text{ack}} + a + \frac{h+s}{r_v \times t_{\text{slot}}})$
      $pdf_{T_{v;u}}[i] \leftarrow pdf_{T_{v;u}}[i]$
                  $+ \Pr\{\alpha_v = a | M_{v;u} = m\}$
                  $\times \Pr\{M_{v;u} = m\} \times \Pr\{\delta_v = c\}$
    end for
  end for
end for
return $pdf_{T_{v;u}}$
```

Table 7.1: Simulation parameters.

Parameter	FourCross	Butterfly
NIC configuration	802.11ac	802.11ac
Node distance ($\mathcal{C}_v$)	≈200 m	≈200 m
Carrier-sense range ($\mathcal{D}_v$)	≈400 m	≈400 m
Frames sent (=sample size)	10000	10000
Data rate	30 Mbit/s	30 Mbit/s
Tx power	3 mW	3 mW

Table 7.2: Rel. constants from IEEE 802.11.

Constant	FourCross	Butterfly
$CW_{\min}, CW_{\max}$	$15, 1023$	$15, 1023$
t_{slot} / t_{ack} / t_{AIFS}	9/18/36 μs	9/18/36 μs
r_{base}	6 Mbit/s	6 Mbit/s
h	320 bit	320 bit
s	1024 bit	1024 bit
$\hat{m}$	7	7

The maximum length of the returned list $pdf_{T_{v;u}}$ is bound by the respective maximum elements from the intervals $a \in [1, Cw(m)]$, $m \in [1, \hat{m}]$, $c \in [0, |\mathcal{D}_v|]$. So the highest slot index is defined as $i_{\max} = Cw(\hat{m}) \times \hat{m} \times |\mathcal{D}_v|$, which is at the same time the worst case latency measured in slot times. The worst case latency $t_{v,u;\max}$ can be determined as

$$t_{v,u;\max} = Cw(\hat{m}) \times \hat{m} \times |\mathcal{D}_v| \times t_{\text{slot}}.$$

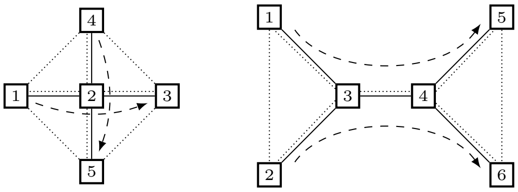
Computing the overall path delay T_P reduces to the convolution of the individual hop's $T_{v;u}$ and the PDF of the queueing delay, as depicted in Algorithm 2.

Algorithm 2 Calculate PDF of $\Pr\{T_P = i \times t_{\text{slot}}\}$ for each slot $i \in \mathbb{N}$

```
$pdf_{T_P} \leftarrow 1, 0, 0, 0, ...$
for all $(v, u) \in P$ do
    $pdf_{T_P} \leftarrow pdf_{T_P} \circledast pdf_{T_{v;u}} \circledast pdf_{Q_v}$
end for
return $pdf_{T_P}$
```

7.4 Model Evaluation Using Simulation

Since we have now modeled all components of the end-to-end delay T_P within IEEE 802.11 WMHN, we can now evaluate it in simulation.



FourCross network

Butterfly network

Figure 7.4: The two simulated networks. The network topology is shown as solid lines, while dotted lines show the conflict graph. Dashed arrows show the paths of haptic flows (a)-(d) noted in Table 7.4.

For evaluation, we have implemented the two networks shown in Figure 7.4 in OMNeT++ and compared latencies with our model prediction. Since our model does not cover queueing delays yet, we can only measure small-scale networks where the utilization is low enough to neglect potential queueing delays. The interference that arises from concurrent transmissions, however, is covered already by the model. The networks chosen in Figure 7.4 respect these constraints. However, we do not expect the measured delays to be distributed exactly, in the statistical sense, as predicted with our model. Instead, we accept our model as valid if it does not underestimate the delay. In other words, all measured delays in the scenarios have to be within the minimum and maximum predicted delays. We hereby consider the 0.999-quantile

Table 7.3: Model parameters.

Parameter	FourCross	Butterfly
r_v	130 Mbit/s	130 Mbit/s
p_{per}	0.009	0.009
p_{cont}	0.25	0.5

Table 7.4: Haptic flows used for the simulation.

Network	Flow	Begin at $t =$	Tx interval
FourCross	(a) $1 \to 2 \to 3$	0 ms	10 ms
FourCross	(b) $4 \to 2 \to 5$	1.5 ms	10 ms
Butterfly	(c) $1 \to 3 \to 4 \to 5$	0 ms	10 ms
Butterfly	(d) $2 \to 3 \to 4 \to 6$	1.5 ms	10 ms

as maximum, since the absolute maximum according to $t_{v,u;\max}$ can result in several hundreds of milliseconds with very unlikely probability.

In each network, there are two pairs of source and destination nodes. The sources send simple UDP messages with 100 B payload, resulting in a 128 B MPDU, at an interval of 10 ms to the respective destinations. The FourCross network shows a simple constellation in which the two flows intersect at a central node that has to relay both flows simultaneously. The Butterfly network is an extension to the FourCross in the sense that the flows now intersect at a common edge between nodes 3 and 4, which adds a conflicting link to share between two flows. We measure 4 flows in total, referred to as (a)-(d), summarized in Table 7.4. Flows (a) and (c) start at simulation time $t = 0$ ms, while (b) and (d) start with a small delay of $t = 1.5$ ms. Each node uses 802.11ac technology with 4x4 MIMO and (potential) 600 Mbit/s physical data rate.

All in all, our model has three model parameters that we have to choose adequately: r_v, p_{per}, and p_{cont}. All other parameters introduced in the previous section are, in fact, constants of the underlying protocol. They do not need to be assumed in their quantity and are taken from the IEEE 802.11 standard. We have summarized their values in Table 7.2. The three model parameters are assumed as follows. The packet error rate p_{per} and data rate r_v were determined in a separate simulation run with a clean environment between two nodes, especially with no interference present. We measured $p_{\text{per}} = 0.009$ and $r_v = 130$ Mbit/s there, which we adopted for our model. We furthermore estimated $p_{\text{cont}} = 0.25$ for the FourCross network, and $p_{\text{cont}} = 0.5$ for the Butterfly network. These pessimistic values stem from the circumstance that the haptic flows in both networks start nearly simultaneously and therefore much interference through contention is to be expected. The model parameters are shown in Table 7.3.

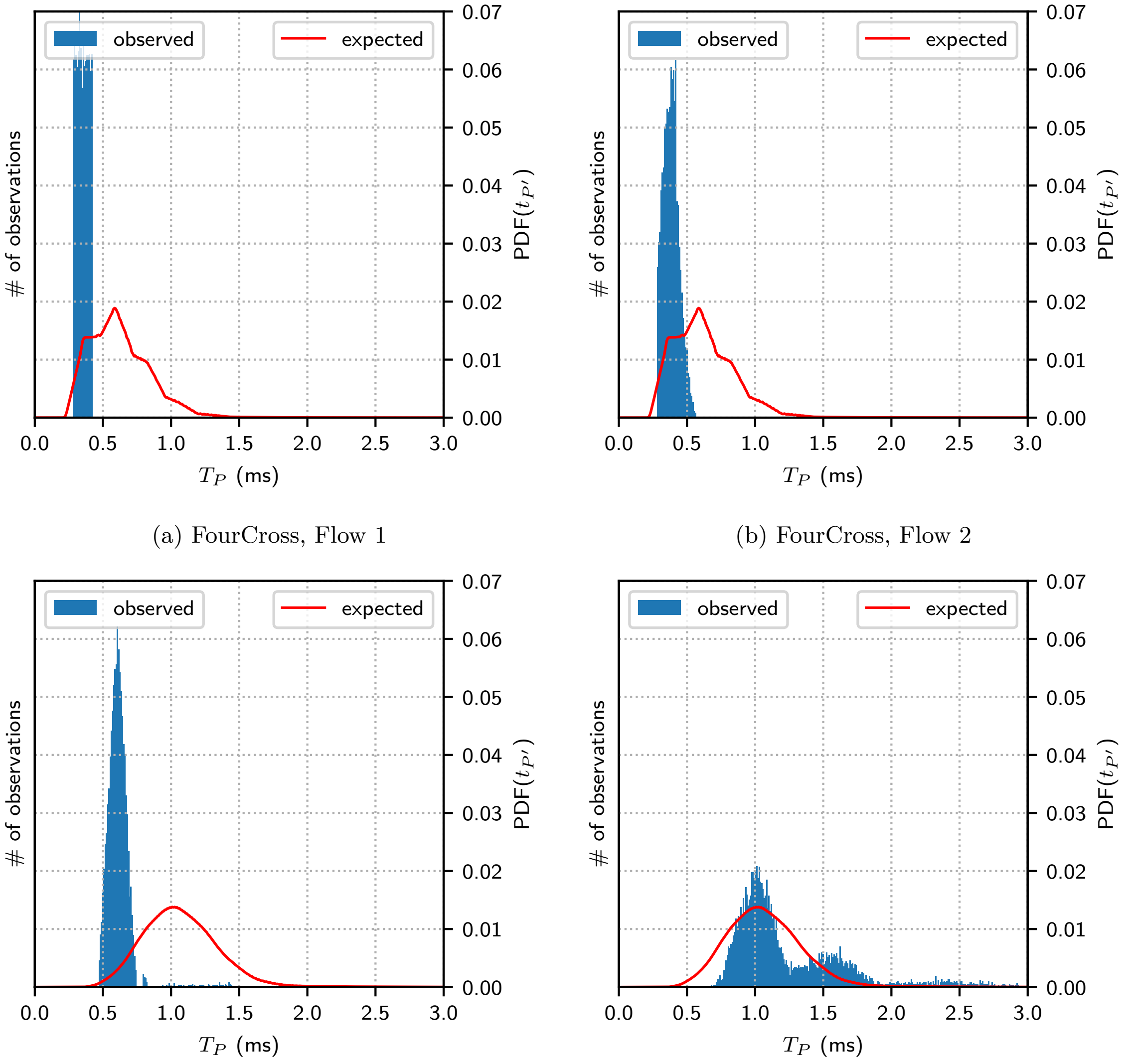
The results are depicted in Figure 7.5 and Table 7.5. The model holds its predictions in terms of minimum and maximum expected delays, which encourages us to adapt and investigate it further. Most notably is the prediction of flow (d) in Figure 7.5, which matches its measured sample distribution surprisingly well.

Table 7.5: Expected and observed results for delay of flows (a)-(d) in our experiments (in ms). Confidence intervals for the observations are according to the central limits theorem with a significance of 0.05.

Flow	0.001-Quantile		0.999-Quantile		1-Quartile		2-Quartile		3-Quartile		Obs. conf. interval	Model σ	Obs. std.-dev
	Exp.	Obs.	Exp.	Obs.	Exp.	Obs.	Exp.	Obs.	Exp.	Obs.			
(a)	0.243	0.283	2.097	0.418	0.459	0.319	0.594	0.346	0.756	0.382	±0.001	0.231	0.043
(b)	0.243	0.283	2.097	0.553	0.459	0.337	0.594	0.382	0.756	0.418	±0.001	0.231	0.059
(c)	0.432	0.474	2.772	1.443	0.873	0.564	1.044	0.609	1.233	0.654	±0.002	0.284	0.106
(d)	0.432	0.700	2.772	4.755	0.873	0.970	1.044	1.094	1.233	1.491	±0.010	0.284	0.508

7.5 Summary and Discussion

In this chapter, we have discussed the specific challenges of latency modeling for the Access Network Plane. We have exercised a probabilistic modeling approach for IEEE 802.11 WMHN with a preliminary model that neglects queueing delays, but only requires three model parameters. In simulations, this model shows already promising in the regard that it does not underestimate the latency in terms of the 0.001-Quantile in two small-scale example



(a) FourCross, Flow 1

(b) FourCross, Flow 2

(c) Butterfly, Flow 1

(d) Butterfly, Flow 2

Figure 7.5: Expected vs. observed latency distributions of flows (a)-(d). Expected distribution is smoothed by a Gaussian kernel of one $t_{\text{slot}} = 9\,\mu\text{s}$ bandwidth for better display.

networks. In the upper ends of the latency distributions, however, the similarity decreases rapidly, and also the expected values and variances do not meet the simulations. This is because the model does not include the network capacity as a parameter. The relatively good matching between expected and observed distributions for experiment (d) seems to indicate a quite high medium utilization within that experiment. Thus, experiments with more flows or longer paths may invalidate the model results, as the queueing delay has more impact and can not be neglected anymore.

The following challenge is, therefore, to include a queueing model to estimate the distribution of the queueing delay as well.

CHAPTER 8

A Matrix-Exponential Queueing Model for Access Networks

In the probabilistic model from the last chapter, we omitted the queueing delay. Queuing delay becomes significant with high medium utilization and is complex in its distribution. Especially true when the conflict graph of a network is dense, the transmission of one node has to block many transmissions from other nodes. The queueing delays of the nodes are significantly interdependent in this case.

In this chapter, we propose a formal queueing model that respects all the relations on the MAC Layer that we have elaborated so far in the Chapters 6 and 7. We call it the ME queueing model, as it is based on the respective class of probability distributions which was investigated deeply by Neuts in the 1970s and is summarized in a book by Lipsky in 2009 [98], from which we widely adopt the notation.

The ME queueing model is based on a conflict graph-based Divide and Conquer model for IEEE 802.11 networks published by Stojanova et al. in a 2018 journal article [100]. This model was originally used to predict the throughput of a non-saturated IEEE 802.11 network, but the approach is useful fur us to model the Matrix-Exponential service times of the queueing model that we have developed in Chapter 6.

The steady-state solution of the *ME / ME / 1* queueing systems we use here is too complex to yield a closed-form analytical solution. Instead, we give an algorithm to sample the PDF numerically. We conclude the chapter with a numerical example.

8.1 Purpose, Prerequisites, and Preconditions

The ME queueing model that we develop during this chapter serves the purpose of modeling the queueing delay in large-scale wireless multi-hop networks. Within the context of this thesis, it serves the modeling part specifically for Access Networks within the Tactile Internet, but can be used apart from the Tactile Internet idea for any arbitrary wireless network. In contrast to state-of-the-art modeling techniques based on Poisson processes (or, more generally, on Gamma distributions) discussed in Section 7.1, it does not rely on a large set of model parameters, but instead deduces the service time distributions of the queueing system directly from the medium access scheme, the channel capacity and the given network load. The relation is expressed in the following Figure 8.1, which shows the input parameters of the model and its purpose in addition to the latency model that we have already developed in Chapter 7. As a side product of this approach, the ME queueing model is more powerful in terms of accuracy, as many protocol-specific effects are taken into account. For example, the effect of spatial reuse (discussed in Section 6.4.1) is included in the model, which allows for a better utilization of channel resources and can only hardly be modeled with Gamma-distributed queueing systems.

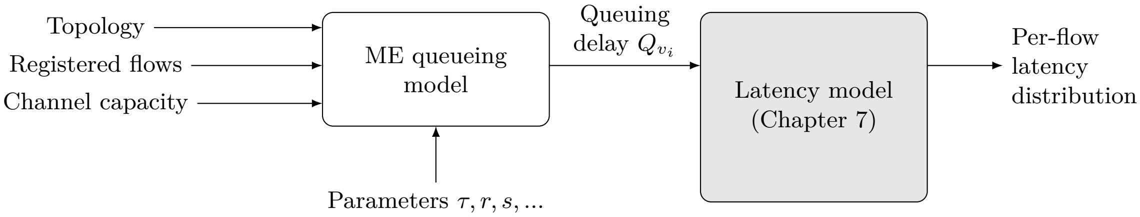

Topology
Registered flows
Channel capacity
ME queueing model
Parameters $\tau, r, s, \ldots$
Queuing delay Q_{v_i}
Latency model (Chapter 7)
Per-flow latency distribution

Figure 8.1: Model inputs and its relation th the latency model developed in Chapter 7.

The ME queuing model itself is based on a set of preconditions. First, we assume that for haptic data a prioritization scheme is present and that haptic flows have the highest priority of all traffic classes. This is a natural prerequisite since the haptic modality has the most stringent QoS and QoE constraints. However, we must also assume that means of enforcement are present and that the priority enforcement is followed rigorously within all parts of the network that are of concern, for example within a whole network segment of an Access Network. More specifically, we assume that the presence of any communication of lower priority (whether it is audio, video, or best effort data), does not affect the latency distribution of any haptic flow. This can be achieved through some means of QoS enforcement, e.g., DiffServ or IntServ. In order to give credit to the feasibility of this assumption, we introduce such a scheme specifically for WiFi networks Chapter 9.

Second, for the sake of simplicity, we assume that each haptic flow has the same QoS constraints. This is a weak prerequisite and can be established easily by selecting the most stringent QoS constraints from all present flows and apply them to all flows equally. It is also a feasible prerequisite, as we have shown in the Background chapter that especially kinesthetic data has very similar shape between different applications. More specifically, all flows within the network segment of consideration are periodic and have the same constant packet rate r and payload size s.

8.2 General Approach

From our formalization in Chapter 6, we have seen that the queueing behavior in the network cannot be modeled in a simple form. Both arrival and service times are of a general distribution, which restricts our toolset for analysis to those rules that apply for *G / G / 1* systems. In such systems, Littles's law is often the only means to express the response time of a queue [99]. It relates the response time W, the queue length N, and the arrival rate λ of a queueing system:

$$\mathrm{E}\left[N\right] = \lambda \mathrm{E}\left[W\right].$$

A distributional version of this law also exists, which enables us to relate the distributions of W and N [116]:

$$N \overset{d}{=} N_a(W).$$

The relation $\overset{d}{=}$ denotes equality in the probability distribution. $N_a(t)$ is a monotonically increasing function that yields the number of customers that have arrived in the system, counting from zero up to time t. $t = 0$ denotes a fresh start of the system with no customers in the queue, so $N_a(0) = 0$. The intuitive interpretation given in [116] is that the number of customers in the queue is equally distributed to the number of arrivals during the waiting time of customers in the queue. Although the monotonic increasing function $N_a(t)$ is an unknown entity to us, as is its inverse, we get a hint on the relation between the response time (or, the queueing delay) W and N.

We use this hint and start modeling the overall queueing system within a whole Access Network segment from the source to the sink, using the relations and the model that we have already developed during Chapter 6. We have to model a whole network segment here, since in wireless networks, each transmission of a single node can potentially affect any other node in the network. A segment here means that we have to consider all nodes that operate on the same frequency within a confined space.

We show that it is possible to translate an entire given network model into a queueing model describing its queueing delay distributions. More specifically, we translate a network with $n_{\mathcal{V}}$ nodes and several n_{f} active flows, with each flow j being associated with a path $\mathcal{P}_j$ of hop-length l_j, into a queueing model that consists of

- $\sum_{j=1}^{f} l_j$ queues,
- each queue having a service comprising of $n_{\mathcal{V}}$ phases,
- and each service being of the same type.

We show that this model is protocol-agnostic, thus applying to both TDMA and CSMA networks. In the former case, the service times will be deterministic if the scheduling is so. In the latter case, it yields a fully Matrix-Exponential (ME) response time distribution.

8.3 Modeling Communication Behavior on the Wireless Channel: The Divide and Conquer Model

We base our node behavior model, which we call ME Queueing Model, on a Divide and Conquer model introduced in a publication from Stojanova et al. [100]. In this section, we first review the Divide and Conquer model. It is a throughput model for non-saturated IEEE 802.11 mesh networks, which we modify by combining it with our queueing analysis. We extend the original Divide and Conquer model by a) giving up the axiom that only CSMA medium access is used by the nodes, b) we embed their medium utilization model into a queuing model, c) we provide a means for determining the input rates at each node by the net of registered haptic flows.

The latency distribution depends on the emergent behavior of the entire set of nodes in the communication channel. When packets are to be transmitted between nodes by means of any (modern) communication protocol, the sequence of channel access must be first negotiated to avoid collisions. The negotiation takes time, and thus requires channel resources. The transmission process can be regarded as the *service process* of a queueing system. The *service* of this queueing system is the successful delivery from a sending buffer to the reception buffer of the next-hop node. If several hops are necessary to reach a destination, a chain of queues emerges, each contributing to an overall state of the entire network. Every successful transmission advances the system state by reducing the queue length at the sending node and increasing it at the receiver node. In other words, by modeling the transmission behavior of all the nodes in the network, we model the service time distribution of the respective queuing system.

We cannot reduce the communication behavior to a mere analysis of overall throughput. First, nodes may not have anything to send over long time periods, and thus do not generate network load at all. This is a distinction of our point of view from most of the exisiting analysis in literature, where throughput could be determined by maximizing some utilization function describing the medium. Such behavior of wireless devices has been studied in literature for a while [101, 102, 107, 109, 111, 117]. Second, path delay accumulates with every hop, so that the analysis can not be extrapolated from the modeling of single-hop links.

8.3.1 Network Load, Sending Regimes and Subnetworks

We begin by summarizing all active input and output connections of a node v and assign them a relative utilization in the range $[0, 1]$. Let x_v be the *relative load* of v, and y_v be the output rate. We say that a node has a relative load $x_v = 1$ when it is desired to send packets with 100 % of the medium capacity, i.e., it has no idle time and is fully saturated with transmissions. The load can exceed 100 %. In this case, the node is overloaded and the input queue grows faster than it is emptied. An admission control system is responsible for avoiding node overload, for example by checking all registered flows and granting only in case of free capacity. Similarly, we define the *relative output rate* y_v of a node, which is the realized amount of busy time (v is busy sending) relative to the channel capacity. It is also called *node utilization*. Obviously, the relation $0 \leq y_v \leq 1$ holds. In general, the relation from x_v to y_v is not trivial, as y_v describes the service process of the sending queue of v, while x_i describes the queue's input rate.

The *channel utilization* (also called *busy fraction*) ρ_v of the channel, as perceived by node v, is defined as

$$\rho_v = y_v + \sum_{u \in \mathcal{D}_v} y_u,$$

the sum of all output rates of nodes within detection range of v. It shows that the output rates y_v are bounded by the channel utilization ρ_v, since $0 \leq \rho_v \leq 1$ must hold as well.

The relative loads can be determined, since all haptic flows are registered, and their QoS-parameters required for Haptic Communication are known.

$$x_v = \frac{1}{\text{capacity}(v)} \sum_{\mathcal{P} \in \mathcal{F}} \sum_{(v,u) \in \mathcal{P}} \text{load}(v)$$

The load (in Bit/s) is determined as described in Section 6.4.1. The function capacity(v) denotes the channel capacity, in Bit/s, as it is at node v. It can be either determined by measurement, or by analytical deduction from the used bandwidth and then analyzing the the *channel budget*. (For a short, introductory summary on the theoretical derivation of the channel capacity, we refer the reader to Appendix 12.0.1.) Our overall goal is to relate the input rate x_v to the output rate y_v by means of a Markov model.

Following Stojanova et al., we define the sending *regime* of a node to be one of $\{\text{On}, \text{Off}\}$, where On means that a node wants to send a packet to one of its neighbors at some given time, while the regime Off means that a node has an empty sending queue. Every node is in regime On for a time fraction of x_v, and in Off for $1 - x_v$, as long as $0 \leq x_v \leq 1$. For a whole network, this means that at some time instant, any fraction of the nodes can be in the On regime, while the other is in the Off regime. The Divide and Conquer approach analyzes these permutations of the network state, referred to as *subnetworks*, individually, in order to reduce the modeling complexity. For each subnetwork $\mathbf{b}_i$, let $b_i(v) \in \{\text{On}, \text{Off}\}$ denote the regime of node v, and let the nodes $v_j \in \mathcal{V}$ be ordered according to their running index j. Then the subnetwork $\mathbf{b}_i$ is defined as a tuple

$$\mathbf{b}_i = (b_i(v_1), b_i(v_2), ..., b_i(v_j), ..., b_i(v_{n_\mathcal{V}}))\,.$$

Without loss of generality, let the subnetworks $\mathbf{b}_i$ be ordered lexicographically according to the relation Off $<$ On, such that node j is active in subnetwork i, iff in the binary representation of i bit j is 1. For example, in a network of three nodes, $\{v_1, v_2, v_3\}$, $\mathbf{b}_5 = (\text{On}, \text{Off}, \text{On})$, and $\mathbf{b}_1 = (\text{On}, \text{Off}, \text{Off})$. $\mathbf{b}_0 = (\text{Off}, \text{Off}, \text{Off})$ denotes the state of the network where no node has any packet to send, and thus all nodes are in Off regime. In contrast, $\mathbf{b}_7 = (\text{On}, \text{On}, \text{On})$ denotes the state that every node wants to send. Let $\mathcal{B} = \{\mathbf{b}_i | 0 \leq i \leq 2^n_\mathcal{V}\}$ be the set of all subnetworks. $\mathcal{B}$ represents the permutation of all possible states of a network $(\mathcal{V}, \mathcal{E})$.

Let now assume that each of the subnets materializes with a certain probability β_i. We can calculate β_i from the probability $p_{i;v;\text{On}}$ that the individual nodes are in the On regime, which can be derived from the input rates x_v.

$$p_{i;v;\text{On}} = \begin{cases} \min(1, x_v) & \text{for } b_i(v) = \text{On} \\ 1 - \min(1, x_v) & \text{otherwise} \end{cases}$$

Then the subnetwork probability calculates as the product of the individual probabilities

$$\beta_i = \prod_{v \in \mathcal{V}} p_{i;v;\mathrm{On}}.$$

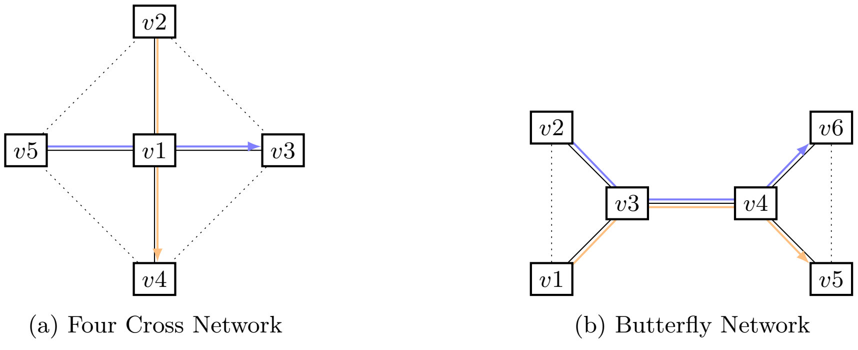
8.3.2 Markov Model for Throughput Analysis

For each of the regimes, we define the sending states and define a Markov model to reflect their relationship. The sending states indicate whether a node is transmitting, whereas the regime only indicates that a packet is ready for transmission in the node's sending queue. For each of the nodes that are in On regime, we now have to consider whether it may be actually sending or not. For each of the possible permutations, we define a *state* (in the sense of classical Markov modeling) of the subnetwork $\mathbf{b}_i$ that indicates which of the nodes are sending.

Let define a Markov chain with the sending states s_k. Let $\mathcal{S} = \{0,1\}^{n_\mathcal{V}}$ be the set of possible sending states for the nodes indexed $v_1, ..., v_{n_\mathcal{V}}$. Each node is represented by a binary state, for which 1 indicates that the corresponding node is actively sending. 0 indicates that the node either is waiting for an idle medium, or is in Off regime. Nodes that are in On regime do not idle without necessity, as the On regime indicates that a packet is in the sending queue. Let $\mathcal{S}_i \subseteq \mathcal{S}$ be the set of states in the Markov chain associated with the subnetwork $\mathbf{b}_i$. Furthermore, let $s_k \in \mathcal{S}_k$ denote the k'th state in Markov chain $\mathcal{S}_k$, and $s_k(v) \in \{0,1\}$ the sending state of node v in the state s_k. Let σ_j^k denote the initialization probability of state s_k in subnetwork $\mathbf{b}_i$.

8.3.3 Examples

At this point, we want to provide two example networks to illustrate the modeling. The Four Cross Network, depicted in Figure 8.2a, consists of five nodes arranged in a cross topology, where one node, v_1 acts as a router between the other four. In the conflict graphs, indicated by dotted edges, the four outer nodes are connected, meaning that they cannot communicate directly, but can also not send at the same time since they are in interference range to each other. Two flows are active in this network, indicated by the blue and yellow unidirectional, two-hop paths. Since both paths cross, the central router acts as the bottleneck in this scenario.



(a) Four Cross Network

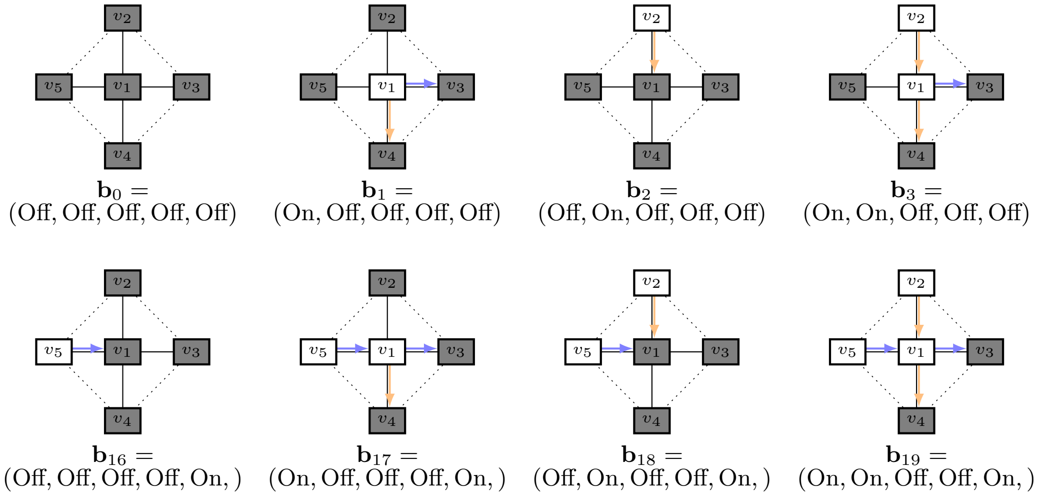
(b) Butterfly Network

Figure 8.2: Example topologies with two flows each.

Similarly, in the Butterfly Network shown in Figure 8.2b, four "leaf" nodes are arranged around central "router" nodes, but this time, a link is shared between the two registered flows, leaving the coordination to the two routers v_3 and v_4.

Figure 8.3 shows the decomposition of the Four Cross network into its subnetworks $\mathbf{b}_i$. Notice that not all nodes do transition into a sending state since not all nodes have packets to forward according to the flow specifications. Thus, not all possible combination from

$\mathbf{b}_0 = (\text{Off}, \text{Off}, \text{Off}, \text{Off}, \text{Off})$ to $\mathbf{b}_{31} = (\text{On}, \text{On}, \text{On}, \text{On}, \text{On})$ are occurring. For example, v_4 only receives data and thus $b_i(v_4)$ never takes the value On.



$\mathbf{b}_0 =$ $(\text{Off}, \text{Off}, \text{Off}, \text{Off}, \text{Off})$

$\mathbf{b}_1 =$ $(\text{On}, \text{Off}, \text{Off}, \text{Off}, \text{Off})$

$\mathbf{b}_2 =$ $(\text{Off}, \text{On}, \text{Off}, \text{Off}, \text{Off})$

$\mathbf{b}_3 =$ $(\text{On}, \text{On}, \text{Off}, \text{Off}, \text{Off})$

$\mathbf{b}_{16} =$ $(\text{Off}, \text{Off}, \text{Off}, \text{Off}, \text{On},)$

$\mathbf{b}_{17} =$ $(\text{On}, \text{Off}, \text{Off}, \text{Off}, \text{On},)$

$\mathbf{b}_{18} =$ $(\text{Off}, \text{On}, \text{Off}, \text{Off}, \text{On},)$

$\mathbf{b}_{19} =$ $(\text{On}, \text{On}, \text{Off}, \text{Off}, \text{On},)$

Figure 8.3: Overview of all subnetworks of the 4cross example.

Let consider the associated Markov chain to $\mathbf{b}_{19} = (\text{On}, \text{On}, \text{Off}, \text{Off}, \text{On})$. We notice that v_1, v_2, v_5 are in On regime and thus may send data, while v_3, v_4 never send data according to in this subnetwork. Furthermore, due to the conflict between v_1, v_2, v_5, only one of them can send at a time. Therefore, the associated Markov chain $\mathcal{S}_i$, shown in Figure 8.4, has only 4 relevant states.

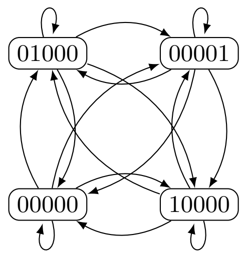


Figure 8.4: Resulting Markov chain for the subnetwork $\mathbf{b}_{19} = (\text{On}, \text{On}, \text{Off}, \text{Off}, \text{On})$ of the Four Cross example.

8.3.4 Transitions and Transition Probabilities

The transitions between sending states now depend basically on the rules of the wireless protocol. The original work from Stojanova et al. determined transition probabilities for CSMA networks, for example WiFi. We also propose transition probabilities for coordinated TDMA networks, as, for example, LTE and 5G. For generality, we thus define the transition probability from state k to state l by means of a MAC-protocol dependent value $\phi_{k;l}$,

$$p_{k;l} = c\phi_{k;l},$$

where c is a normalizing constant. c is chosen such that $\sum_{l \in \mathcal{S}_i} p_{k;l} = 1$ for each k.

For TDMA Networks

TDMA networks do not have any restrictions in terms of state transitions. Through their global coordination, multiple nodes can change their transition state at the same time. In the general case, they are also not prioritized in the likelihood of their transition, but can be if necessary. We, therefore, assume equally likely, constant transition probabilities $p_{k;l}$ from state k to state l in the subnetwork i. Self-transitions are allowed, as well as any transition to another sending state.

$$\phi_{k;l} = \frac{1}{|\mathcal{S}_i|}.$$

The initial probabilities σ_k^i are similarly defined as

$$\sigma_k^i = \frac{1}{|\mathcal{S}_i|}$$

to reflect a uniform initial distribution.

For CSMA Networks

Stojanova et al. introduced a rule for CSMA networks to reflect their relatively narrow transition set. In such a network, they only allow the transition from 0 to 1 for one node at a time, and also only one node can transition from 1 to 0. The transitions are to be made between neighboring nodes. As the nodes are not coordinated as in TDMA, the probability that two nodes change their state at the same time is small. As a result, the set of allowed transitions is reduced, and thus the CSMA networks are less flexible than the TDMA-based networks.

The transition probability $p_{k;l}$ from state k to state l in a CSMA-based network is given by Stojanova et al. by

$$\phi_{kl} = \prod_{v|s_l(v)=1} \frac{1}{1 + \sum_{u \in \mathcal{W}_u} \delta(b_i(u), \text{Off})}.$$

Here, $\mathcal{W}_u = \{v \in \mathcal{D}_u \backslash \{u\} | u$ is not blocked in s_l by a node in $\mathcal{D}_v \backslash \{u\}\}$ is the reduced neighborhood of u that contains all nodes in u's neighborhood that are not blocked by another sending node. This restriction implements the above CSMA rule and de-harmonizes the transition probabilities.

8.3.5 Steady State Probabilities

For analyzing the throughput, we are interested in the long-term behavior of the network, and thus we calculate the steady state probabilities using standard approaches. We again follow the approach from Stojanova et al. The resulting Markov chains for each subnetwork are split up into their irreducible subchains, and weighted with a factor ω_i^j, where i denotes the subnetwork ($\mathbf{b}_i$) and the index j counts the irreducible Markov chains of the subnetwork $\mathbf{b}_i$. The set of states of the Markov chain j of subnetwork $\mathbf{b}_i$ be denoted as $\mathcal{S}_i^j$. This set contains all sending states that form an irreducible Markov chain for the corresponding subnetwork. The weighting factor ω_i^j is then determined by the set of paths that can reach the irreducible subnetwork j. The sum $\sum_j \omega_i^j$ must equal 1. The subchain j's steady state probability π_j^i.

8.3.6 Calculation of the Mean Output Rates

With the now complete Divide and conquer Model, the (mean) output rate of each node v can be calculated according to the formula derived by Stojanova et al. [100]:

$$y_v = \sum_{i=1}^{|\mathcal{B}|} \left(\delta(b_i(v), \text{On}) \times \beta_i \times \sum_{j=1}^{z_i} \left(\omega_i^j \times \sum_{k|s_k \in \mathcal{S}_i^j} \left(\delta(s_k(v), 1) \times \pi_j^i(k) \right) \right) \right)$$

It can be used to calculate the mean latency with the classic version of Little's formula. However, we can not derive the underlying distribution and thus it is only of limited use. In the next step, we utilize the throughput model from the Divide and Conquer Approach to obtain a service distribution for a queueing model, as we develop in the following Section.

8.4 Modeling as Matrix-Exponential Queueing Processes

In this section, we refine the Divide and Conquer to a queueing model. We borrow the Markov model that represents the channel access and use it as a service process for a queuing system that represents the packet flow along a path (or multiple paths) in the network. The way we embed it as a service process for a queue is known as Matrix-Exponential (ME) queuing system in literature. We start with an introduction to this class of queuing systems before we describe our model. We evaluate the resulting ME queuing model analytically in two example networks.

8.4.1 Matrix-Exponential Queueing Processes

In the common queueing analysis, both the server and the arrival processes are modeled using a trivial (unimodal) probability distribution, and among these, the exponential process model (also called Markov process or Poisson process) is the most widely used. Unimodal distributions can be described with a limited set of parameters. The exponential distribution, for example, has a PDF of the form

$$f_T\left(t\right) = \lambda \mathrm{e}^{-\lambda t}$$

and is fully specified by the parameter λ, which can be interpreted as a rate of incoming events. The probability density function gives notice about the probability of the inter-arrival time between two events. In a queuing system of the form *M / M / 1*, for example, both the arrivals and service times are distributed in that manner, which forms a well-understood behavior.

Matrix-Exponential (ME) queueing systems are another class of queueing systems that allow for more complex behavior than trivial queueing systems, but still, allow for deeper analysis beyond Little's Law. They allow for a specification of a system behavior that can be described by a Markov model, which means that the service process (or, similarly, the arrival process) can be modeled using a Markov chain. The resulting service time distribution (or, arrival distribution) can thus be multimodal, and may not be described by a trivial distribution function, such as the Gamma distribution. In fact, the existence of a closed-form solution for general ME queueing systems is still unknown [98]. However, they help to model and analyze the latency distribution since they are able to map the complex process of channel access that we have modeled in the previous section. They have been investigated thoroughly by Neuts [118]. In this thesis, we follow the notation of Lipsky [98].

With ME queuing systems, the server (or the arrival process) consists of a set of internal states, referred to as *phases*, to avoid notational confusion. If a customer (i.e. an event) arrives at the system, it enters it in a phase that is randomly chosen. It then traverses through the subsystem by a rule set similar to that of Markov processes. But unlike with Markov processes, it can leave the subsystem if it enters a phase that has a nonzero exit probability. If the customer leaves the subsystem, it is considered as “processed” by the subsystem, and a new customer enters the subsystem immediately if the is another one in the queue. As with other queuing systems, a maximum of one customer can be processed at any time. Multiplication of the server is, however, possible and is then referred to as *ME / ME / N* system.

Figure 8.5 shows an example of such a system. It has four phases. When a customer enters the system, it enters one of the four phases s_1, s_2, s_3 with probability p_1, p_2, p_3, respectively. It can then randomly traverse through the phases with given transition probabilities $p_{k;l}$.

Subsystems don't need to be loop-free, as in this example, so a customer may take and indefinite amount of time to exit the system. The probability distribution of the overall

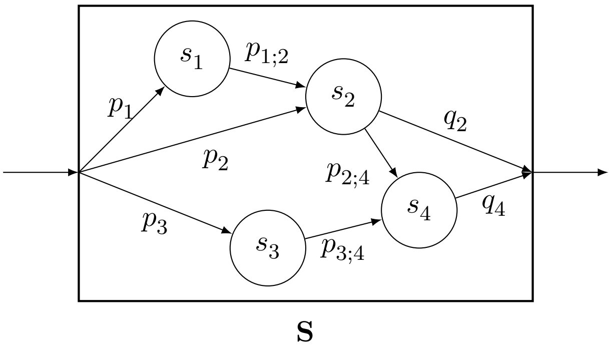


Figure 8.5: A non-exponential subsystem consisting of $o = 4$ phases.

service time results from the absorption behavior of the Markov Chain that is formed by the phases.

Let us now introduce a formalization for ME systems [98]. Let o be the number of phases in the subsystem. The vector of initial probabilities $\mathbf{p} = [p_l]^{1\times o}, p_l \in [0,1]$, describes with which probability a phase is initially chose upon subsystem entry. The transition matrix $\mathbf{P} = \left[p_{k;l}\right]^{o\times o}$ describes the probability of a transition from phase k to phase l. The exit vector $\mathbf{q} = [q_l]^{1\times o}, q_l \in [0,1]$, describes the probability that the subsystem is left from a phase l. As for a probability distribution, it holds that for each $l, 1 \leq k \leq o$

$$q_k + \sum_{l=1}^{o} p_{k;l} = 1,$$

and thus the transition matrix and the exit vector completely specify the transition behavior within the subsystem. Similarly,

$$\sum_{l=1}^{o} p_l = 1.$$

The semantics of a ME type subsystem can be described by that of a Markov chain that is obtained as follows: The set of states is the set of ME phases, plus an extra, unique absorbing state $\mathbf{q}$. This absorbing state acts as a recipient for all transitions that are meant to exit the ME subsystem. The initial probabilities of the Markov Chain are defined by the vector $\mathbf{p}$ ($\mathbf{q}$ is assigned the initial probability 0). The transition probabilities of the Markov Chain are obtained by extension of $\mathbf{P}$ by $\mathbf{q}$, where $\mathbf{q}$ receives all transitions described by $\mathbf{q}$. The further analysis is then concerned about the absorption behavior of the state $\mathbf{q}$.

Both *M / ME / N* and *ME / M / N* can be analyzed analytically in a feasible way [98]. In a *ME / ME / 1* queueing system, both the arrival and the service process are of ME type, and such systems, to the day, can not be analyzed analytically. Here only a numerical approach can be taken for obtaining probability distributions.

8.4.2 Representation of the Network Transmission Process

To adapt the ME queueing model to the network transmission process, we define a chain of queues, each having a ME type service process, as depicted in the following Figure 8.6. Assuming that a flow $\mathcal{F}$ is reserved to transmit haptic data along the path v_1, v_2, v_3, v_4, we model each hop in this transmission chain with such a queueing process. Attached queues serve as buffers for packets, where the service processes, here denoted with $\mathbf{S}^{(v_1)}, \mathbf{S}^{(v_2)}$, and $\mathbf{S}^{(v_3)}$, determine the channel access of the respective nodes. In this example, packets generated by the source host v_1 will be placed in the queue attached to v_1. Packets have to be processed in the respective service process to enter the respective process on the next node, so the service process $\mathbf{S}^{(v_1)}$ takes the first packet from its queue, which represents the process of channel access in the intention to transmit the packet. A packet is eliminated when it completes the

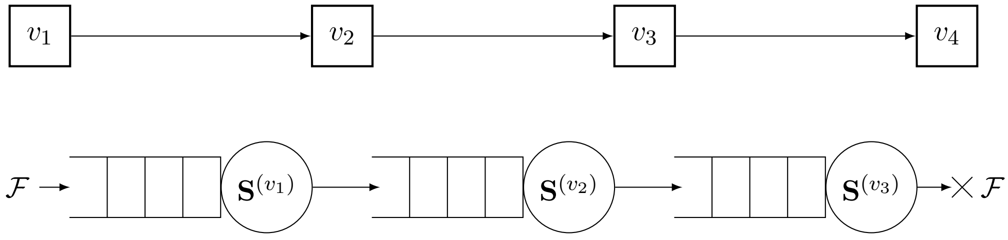


Figure 8.6: Queueing model for a linear path over three hops.

last service process in the chain and thus passes beyond the last hop, in which case it reached its destination host. The × symbol after the last service process indicates this termination process.

The packet generation process (on the left side of Figure 8.6) follows the rules that we have introduced in Section 8.1 in this chapter. It is an isochronous creation process that generates packets at a constant rate. The first queueing system, therefore, formally is of *D / ME / 1* type, while all other queueing systems are of *ME / ME / 1* type.

We are now concerned with the analysis of the distribution of the lifetimes of packets in this system. This distribution is by definition identical with the distribution of the packet latency. To do this, we have to formally define the service processes $\mathbf{S}^{(\cdot)}$. For notational clarity, we follow the notation of Lipsky and refer to ME type service processes as *subsystems*.

8.4.3 Formalization of Subsystems in the ME Queuing Model

The subsystems in the aforementioned chain of queues represent the process of channel access at the given nodes in the network. We have already sketched this process by the Divide and Conquer model presented earlier. We now merge the two models by formally defining the semantics of a subsystem $\mathbf{S}^{(v)}$ corresponding to a node v.

A subsystem is defined as the tuple $\mathbf{S}^{(v)} = \big(\mathcal{Q}, \mathbf{p}^{(v)}, \mathbf{P}^{(v)}, \mathbf{q}^{(v)}\big)$, where $\mathcal{Q} = \mathcal{B} \times \{0,1\}^{n}_{\mathcal{V}}$ is the set of phases, $\mathbf{p}^{(v)} \in [0,1]^{1\times o}$ is the initialization vector, $\mathbf{P}^{(v)} \in [0,1]^{o\times o}$ is the matrix of transition probabilities, and $\mathbf{q}^{(v)} \in [0,1]^{1\times o}$ is the vector of phase exit probabilities. The number of phases o in a subsystem calculates as

$$o = |\mathcal{Q}| = 2^{2n_{\mathcal{V}}}.$$

The set of phases $\mathcal{Q}$ enumerates all combinations of subnetworks (where nodes are classified into either On or Off regimes) and node sending states (where nodes in On regime can either send or idle/receive, denoted by a bit 1/0). Note that not all tuples in $\mathcal{Q}$ thus represent a valid network state, as nodes in Off regime may be assigned a sending state 1. These phases, however, will be entered with probability 0, and thus do not contribute to the subsystem's behavior. Thus, we are only interested in phases that have a nonzero probability of being reached. We call reduced set of valid phases $\mathcal{Q}'$, which is defined as follows. First, $\mathcal{Q}'$ shold not contain phases where nodes are marked to be in in sending state ($s_g = 1$), but are actually in Off regime. Second, the conflict graph may forbid nodes to be sending simultaneously, which should also be excluded from $\mathcal{Q}'$. Thus,

$$\begin{aligned} \mathcal{Q}' = \mathcal{Q} \quad &\setminus \quad \{(\mathbf{b}_i, s_g) \in \mathcal{Q} \mid \exists v \in \mathcal{V} : b_i(v) = \text{Off} \land s_g(v) = 1\} \\ &\setminus \quad \{(\mathbf{b}_i, s_g) \in \mathcal{Q} \mid \exists (u,v) \in \mathcal{E}_{\text{con}} : s_g(u) = 1 \land s_g(v) = 1\} \end{aligned}$$

The actual number of valid phases $o' = |\mathcal{Q}'|$ is, therefore, much lower than o.

As described earlier, the initial probabilities $\mathbf{p}^{(v)}$ define the likelihood that a phase k is chosen upon subsystem entry. Since a subsystem $\mathbf{S}^{(v)}$ is entered if and only if a packet has arrived in the sending queue of the corresponding node v, v must be in the On regime throughout the whole service time of the subsystem. This means, all phases that can be entered with probability ≥ 0 must be in On regime. For these phases, we have already derived the initial probabilities in Section 8.3 as the product of the initial probabilitiy corresponding to the subnetwork and the initial probability of the specific state within that subnetwork. We apply this rule here, too, and adjust these values according to the node's regimes. Thus, the initial probabilities $\mathbf{p}^{(v)} = \left[p_k^{(v)}\right]$ for each phase $k = (\mathbf{b}_i, s_g) \in \mathcal{Q}$ are defined as

$$p_k^{(v)} = \begin{cases} \hat{c}_k^{(v)} \beta_i \sigma_g^i & \text{for } k \in \mathcal{Q}' \text{ and } b_i(v) = \text{On} \\ 0 & \text{otherwise.} \end{cases}$$

σ_g^i, as in Section 8.3 is the initialization probability of state sending state s_g in the subnetwork $\mathbf{b}_i$. $\hat{c}_k^{(v)}$ is a correction factor to enforce the property $\sum_{k \in \mathcal{Q}} p_k^{(v)} = 1$, since we force initial probabilities of states where v is in Off regime to zero.

$$\hat{c}_k^{(v)} = \frac{1}{\sum_{(\mathbf{b}_j, s_h) \in \mathcal{Q}'} \beta_j \sigma_h^j \delta(b_j(v), \text{On})}$$

When the node v is in sending state 1, it reduces its buffer by one packet, and thus the subsystem $\mathbf{S}^{(v)}$ should exit – the service process has successfully processed a packet in the queue. If not, the exit probability is zero, as a packet can leave the system only through a successful transmission. Let phase $k = (\mathbf{b}_i, s_g) \in \mathcal{Q}$, then the exit probability $q_k^{(v)}$ for k is

$$q_k^{(v)} = \begin{cases} 1 & \text{for } s_g(v) = 1 \\ 0 & \text{otherwise} \end{cases}$$

The transition probability $p_{k;w}^{(v)}$ from a phase k to a phase w (within the subsystem $\mathbf{S}^{(v)}$ corresponding to node v) is as well defined similarly to the Divide and Conquer model from Section 8.3. The major difference is that a subsystem is now representing an open, recurring system that is left as soon as the current packet has been successfully transmitted. The phase transitions (e.g. the subnetwork and sending state transitions) are iterated through until subsystem $\mathbf{S}^{(v)}$ has successfully serviced the current packet. Unlike with the Divide and Conquer model, we have to include a transition between subnetworks, i.e., we have to allow nodes to alter their regimes from On to Off, or vice versa. To do this, we introduce a parameter τ that indicates the probability that a phase transition involves the alternation of the sending state of some node $u \neq v$ within the network. We do not allow v to change its regime within the subnetwork $\mathbf{b}_v$, as an active $\mathbf{b}_v$ always indicates that a packet is in v's transmission queue, and thus its regime is On. All in all, the transition probability $p_{k;w}^{(v)}$ from phase $k = (\mathbf{b}_i, s_g) \in \mathcal{Q}$ to phase $w = (\mathbf{b}_j, s_h) \in \mathcal{Q}$ is defined as follows

$$p_{k;w}^{(v)} = \begin{cases} 0 & \text{if } s_g(v) = 1 \\ 1 & \text{if } s_g(v) = 0 \text{ and } k = w \text{ and } k, w \notin \mathcal{Q}' \\ 0 & \text{if } s_g(v) = 0 \text{ and } k \neq w \text{ and } w \notin \mathcal{Q}' \\ \tau c_k^{(v)} \phi_{s;g} + (1-\tau) p_k^{(v)} & \text{if } s_g(v) = 0 \text{ and } w \in \mathcal{Q}' \text{ and } i = j \\ (1-\tau) p_k^{(v)} & \text{if } s_g(v) = 0 \text{ and } w \in \mathcal{Q}' \text{ and } i \neq j \end{cases}$$

$\phi_{s;g}$ is the transition probability between sending states s and g as defined in the preceding Section 8.3.4. τ is a parameter indicating the proclivity to change between subnetworks, i.e., between sending regimes of the nodes. $c_k^{(v)}$ is again a normalizing constant to ensure the probabilities sum up to 1, i.e.

$$1 = q_k^{(v)} + \sum_{w \in \mathcal{Q}} p_{k;w} \text{ for each } k \in \mathcal{Q}.$$

Notice that if a phase represents that the corresponding node v is sending ($s_g(v) = 1$), it has no outgoing transition to another phase, but an exit transition with probability 1. The relation $i \neq j$ in the above formula means that a phase transition references a change in the regime in one (or more) nodes in the network, where $i = j$ means that the regimes do not change. The parameter τ controls this behavior.

The service process, or subsystem, $\mathbf{S}^{(v)}$ for a node v is now fully defined for the ME queueing model, and analysis for linear networks as shown in Figure 8.6 is possible.

8.4.4 Service and Waiting Time Distributions, Number of Packets in the Queue

The service time distribution of a subsystem $\mathbf{S}^{(v)}$ can now be determined by the vectors $\mathbf{p}^{(v)}, \mathbf{q}^{(v)}$, and the matrix $\mathbf{P}^{(v)}$. Let $\pi_k^{(v)}(t)$ be the probability that the subsystem $\mathbf{S}^{(v)}$ is in phase k at time step t. The probability vector $\pi^{(v)}(t) \in [0,1]^{1\times o}$, $\pi^{(v)}(t) = \left[\pi_1^{(v)}(t), \pi_2^{(v)}(t), \dots, \pi_o^{(v)}(t)\right]$, can be determined as [98]

$$\pi^{(v)}(t) = \mathbf{p}^{(v)} \, \mathbf{P}^{(v)\,t} .$$

The service time distribution is defined by the inverse sum over all components of $\pi^{(v)}(t)$. Let S_v be the random service time of the subsystem $\mathbf{S}^{(v)}$, then the cumulated distribution function $F_{S_v}(t)$ is

$$F_{S_v}(t) = 1 - \pi^{(v)}(t)\epsilon,$$

from which we can derive the probability density function $f_{S_v}(t)$ by differentiation. $\epsilon = [1, \dots, 1]^T$ is the column vector consisting of all ones.

It should be noted here that the term $\pi^{(v)}(t)\epsilon$, in contrast to classical Markov chains, does not compute to 1 for Matrix-Exponential subsystems. Since the exit vector $\mathbf{q}^{(v)}$ has nonzero components for phases that can be reached during the transmission process, the chance for transmission, and thus for the terminating subsystem, increases with each time slot, finally approaching 1 for $t \to \infty$.

For the waiting time Q_v, we can then convolve the service time density function with itself according to the number of packets N_v in the queue. Let $f_{Q_v|N_v}(t|n)$ be the probability density function of the waiting time, given that n packets are waiting in the queue. It is defined recursively by

$$f_{Q_v|N_v}(t|n) = \begin{cases} f_{S_v}(t) & \text{for } n = 0 \\ \int_{-\infty}^{\infty} f_{Q_v|N_v}(t|n-1)\, f_{S_v}(t-s)\,\mathrm{d}s & \text{otherwise.} \end{cases}$$

Concatenation of Queues, and Expected Queue Lengths

The above ME model yields a complete specification for the individual queue's service processes. However, we also have to specify how the queues are interconnected. Since flows origin from and end in arbitrary nodes, the packets belonging to distinct flows are routed individually. Therefore, we need to define *splitting* and *joining* operations.

A joining operation occurs in situations as depicted in Figure 8.7. Here, two, or more flows join at a node v. This is the case in situations where the flow paths intersect at that node v, so v acts as an intermediate node in all of the flows.

For each joining operation, the outbound packet rate is the sum of the packet rates of all inbound flows. As a result, there is a negative effect on the queue length at v, as v mus serve all the joining flows.

In a splitting operation, as depicted in Figure 8.8, the path of two flow (or of several flows) split up towards distinct nodes.

Similarly to joining operations, the sum of the individual packet rates of all outgoing flows is the same as the inbound packet rate.

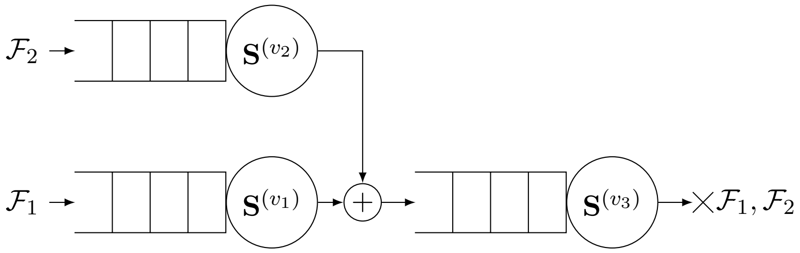


Figure 8.7: Two flows joining into one queue.

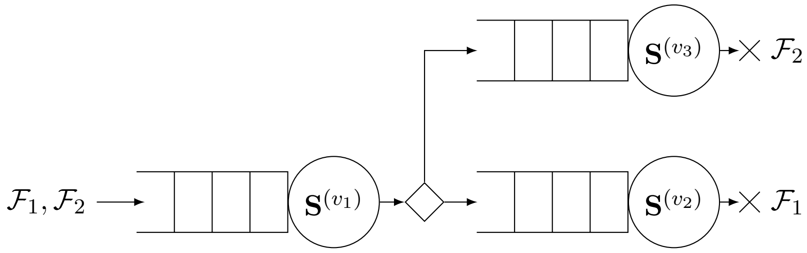


Figure 8.8: Two flows splitting up into two different queues.

We estimate the number of waiting packets N_{v_i} in the queue of node v_i by their expected values from Littles's Law and the approximation for the output rate y_{v_i}:

$$y_{v_i} = \frac{1}{\mathrm{E}\left[S_{v_i}\right]},$$

and

$$\mathrm{E}\left[N_{v_i}\right] = \frac{1}{y_{v_i} - x_{v_i}}.$$

y_{v_i} is the service rate of the transmission process of node v_i.

8.4.5 Resulting Algorithm for Determining the Waiting Time Distributions Q_v

As a result of the above elaborations, we have now the tools to fully model the waiting time distribution $f_{Q_v}(t)$. We can now summarize our approach in form of an algorithm. The idea is to first model the network with the formal approach given in Sections 8.3, where we construct the individual ME service process models $\mathbf{S}^{(v_i)} = \left(\mathcal{Q}, \mathbf{p}^{(v)}, \mathbf{P}^{(v_i)}, \mathbf{q}^{(v_i)}\right)$ for each node v_i, and then determine the distributions of Q_{v_i} and S_{v_i}. However, the formulae from the ME model are far too complex for analytical analysis. We have to approximate the solution numerically by sampling the resulting PDFs of $f_{S_{v_i}}(t)$ and $f_{Q_{v_i}}(t)$ for each $t > 0$ up to a certain limit $t_{\max}$.

In our numerical solution given in Algorithm 3, we first sample the distribution $f_{S_{v_i}}(t)$ and then determine Q_{v_i} via the known conditional relation $\Pr\left\{Q_{v_i} | N_{v_i} = n\right\}$. We approximate n with the expected queue lengths $\mathrm{E}\left[N_v\right]$ for each node v_i's queue using Littles Law. Finally, we can then sample $f_{Q_v|N_v}\left(t|\mathrm{E}\left[N_v\right]\right)$ as an approximation of $f_{Q_v}(t)$.

The PDFs are stored as lists of floating point values, where each list entry represents the probability at a discrete transmission time slot t, $t \in \mathbb{N}, 0 \leq t \leq t_{\max}$. Assuming that we want to sample a total of $n_{\text{slots}} = t_{\max} + 1$ time slots, the runtime of this algorithm is

$$\mathcal{O}\left(\left(o'^2 \times {n_{\text{slots}}}^2\right) \times n_{\mathcal{V}}\right).$$

o' is the number of valid phases, $n_{\mathcal{V}}$ is the number of nodes in the graph.

The approximation $f_{Q_{v_i}}(t) \approx \Pr\left\{Q_{v_i} = t | N_{v_i} = \mathrm{E}\left[N_{v_i}\right]\right\}$ can be precised if an assumption of the distribution of the queue length N_{v_i} can be made. For example, if N_{v_i} can be assumed to be geometrically distributed (as is for *M / M / 1* queues), a numerical approximation can

Algorithm 3 Sample all PDFs of Q_{v_i} for $v_i \in \mathcal{V}$ from $t = 0$ to $t_{\max}$

```
Initialize all $pdf_{Q_{v_i}}$ for all $v_i \in \mathcal{V}$ to $[0, 0, 0, ...]$
Determine the set of valid phases $\mathcal{Q}'$
for all $k \in \mathcal{Q}'$ do
  Calculate initial probability $p_k$ for phase $k$
end for
for all $v_i \in \mathcal{V}$ do
  for all $k \in \mathcal{Q}'$ do
    Calculate exit probability $q_k^{(v_i)}$ from phase $k$
    for all $w \in \mathcal{Q}'$ do
      Calculate transition probability $p_{k;w}^{(v_i)}$ from phase $k$ to $w$
    end for
  end for
  Initialize $pdf_{S_{v_i}}$ to $[0, 0, 0, ...]$
  Initialize $\pi$ to $\mathbf{p}$
  Initialize $last$ to 0.0
  for all $t \in \mathbb{N}, 0 \leq t \leq t_{\max}$ do
    $pdf_{S_{v_i}}[t] := 1 - \pi\epsilon - last$
    $last := 1 - \pi\epsilon$
    $\pi := \pi\mathbf{P}^{(v_i)}$
  end for
  Calculate $\mathrm{E}[N_{v_i}]$
  Initialize $pdf_{Q_{v_i}}$ to $[1, 0, 0, ...]$
  for all $n \in \mathbb{N}, 0 \leq n \leq \mathrm{E}[N_{v_i}]$ do
    $pdf_{Q_{v_i}}$ := convolution of $pdf_{Q_{v_i}}$ with $pdf_{S_{v_i}}$
  end for
end for
return all lists $pdf_{Q_{v_i}}$
```

be found by sampling $\Pr\{Q_{v_i} = t | N_{v_i} = n\}$ up to a certain upper bound $n_{\max}$.

$$f_{Q_v}(t) \approx \sum_{n=0}^{n_{\max}} \Pr\{N_{v_i} = n\} \times f_{Q_v|N_v}(t|n)$$

8.5 Numerical Results

To illustrate the modeling and the analysis, we want to give an example for the Four Cross Network as shown in Figure 8.2a. The network consists of five nodes, $v_1, ..., v_5$, which host two flows, $\mathcal{F}_1$ on the path $v_5 \to v_1 \to v_3$ and $\mathcal{F}_2$ on the path $v_2 \to v_1 \to v_4$. Figure 8.9 shows the queueing model for this network. Notice that both v_3 and v_4 are not modeled at all, since they only receive packets and thus do not have transmission queues. Only v_2, v_5, and v_1 appear in the model. v_1 acts as a router, while v_2 and v_5 are source hosts of some application. The joining operation (+) in the center indicates the role of v_1 as a router where the two flows cross each other. As soon as v_1's service is completed for a packet from either of the two flows, the respective destination host has been reached and the flows are terminated (×).

The subsystems $\mathbf{S}^{(v_2)}$, $\mathbf{S}^{(v_5)}$, $\mathbf{S}^{(v_1)}$ model the channel access at the respective nodes by means of their internal phase arrangement. This phase model has to map the channel contention, which is fairly complex even for this little example: The source nodes v_2 and v_5 are conflicting with each other, so they can not send at the same time, although they are not connected through a common flow path. Similarly, v_1 conflicts with all other nodes. In this situation,

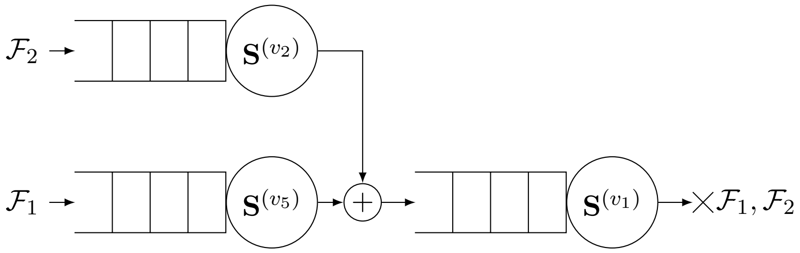


Figure 8.9: Queueing model for the Four Cross Network.

Table 8.1: Parameter values for numerical analysis of the Four Cross Network.

Parameter	Value	Parameter	Value
τ	0.9	$\mathcal{V}$	$\{v_1, v_2, v_3, v_4, v_5\}$
x_{v_1}	0.3	$\mathcal{E}$	$\{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5)\}$
x_{v_2}	0.15	$\mathcal{E}_{\text{con}}$	$\mathcal{E} \cup \{(v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_2)\}$
x_{v_3}	0	$\mathcal{P}_{\mathcal{F}_1}$	$\{(v_5, v_1), (v_1, v_3)\}$
x_{v_4}	0	$\mathcal{P}_{\mathcal{F}_2}$	$\{(v_2, v_1), (v_1, v_4)\}$
x_{v_5}	0.1		

only one of the three active nodes can send at a given time. This exclusivity behavior is modeled by the subnetwork relations within the three subsystems $\mathbf{S}^{(v_2)}$, $\mathbf{S}^{(v_5)}$, and $\mathbf{S}^{(v_1)}$.

To have an insight into the correct behavior of the model in this regard, the subsystem $\mathbf{S}^{(v_1)}$ is shown in detail in Figure 8.10. The subnetworks $\mathbf{b}_0$ to $\mathbf{b}_{19}$ shown are the eight valid subnetworks for the scenario, representing the valid permutation of On/Off regimes that respect the conflict graph and the flow load (i.e., v_3 and v_4 are never in On regime). This results in a set $\mathcal{Q}'$ of 20 phases representing the valid activation states of the nodes within their respective subnetworks. In the figure, the phases belonging to a subnetwork are ordered from top to bottom. Each phase is labeled with its corresponding tuple $k = (\mathbf{b}_i, s_g) \in \mathcal{Q}$ ($i \in \{0, 1, 2, 3, 16, 17, 18, 19\}$ and g being a binary representation of the respective activity state of the nodes in that subnetwork). For example, the phase label $\mathbf{b}_{19}00010$ represents the phase for subnetwork $\mathbf{b}_{19} = (\text{On}, \text{Off}, \text{Off}, \text{On}, \text{On})$ (i.e., all three nodes v_1, v_2, v_5 have a packet in their transmission queue), and only the node v_2 is actively sending.

It can be seen that the restrictions from the conflict graph are respected in the subsystem, as in this scenario no two nodes can send at the same time. This is indicated by the fact that in each phase, either exactly one node is active, or no node at all. The model is, however, powerful enough to support the activation of several nodes at the same time, given that the network is large enough so that spatial reuse can be modeled with it.

8.5.1 Initialization and Exit Vectors, Transition Matrix

The transition probabilities $p_{k;w}^{(v_i)}$, the initial probabilities $p_k^{(v_i)}$, and the exit probabilities $q_k^{(v_i)}$ can be calculated according to the formulae in Section 8.4. The first step would be to assess (or measure) the network capacities capacity(v_i) for each node. Then, we have to calculate load(v_i) based on the flow parameters as packet rate, packet size, etc., to calculate the relative loads x_{v_i} as discussed in Section 8.3.1. We shorten this initial assessment of the relative loads and define the values x_{v_i} for our example as follows. We assume that each flow utilizes 15 % of each node's capacity, and the capacities of each node to be an equal amount. Since both v_5 and v_2 are a source of one of the two flows, their relative loads x_{v_2}, x_{v_5} must be 0.15. For v_1, where the two flows cross each other, the relative load must be twice as much, thus $x_{v_1} = 0.3$. Table 8.1 summarizes all parameters used for the model, including the paths of the two flows $\mathcal{F}_1, \mathcal{F}_2$, which were introduced in Figure 7.4a.

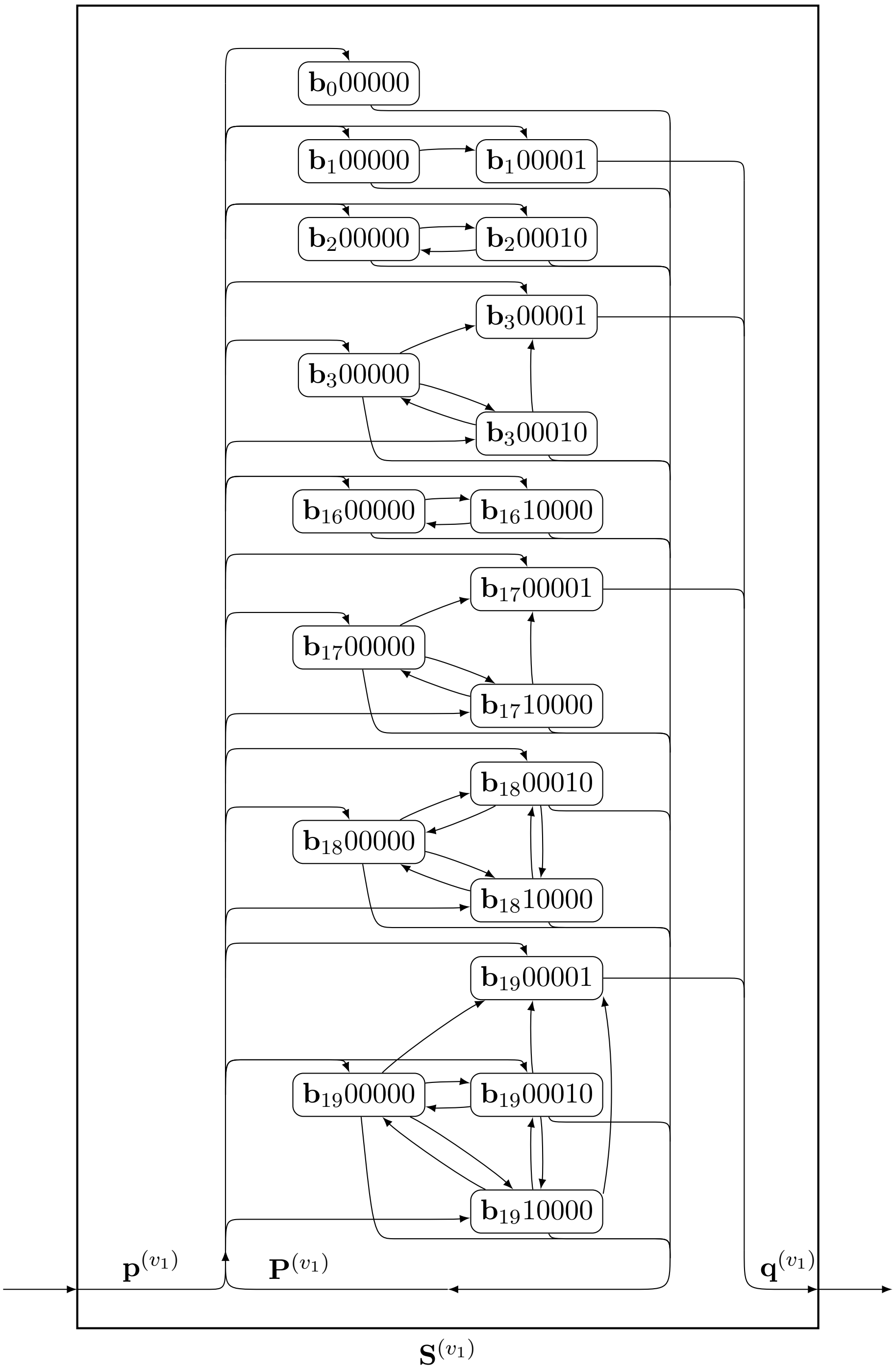


Figure 8.10: Subsystem $\mathbf{S}^{(v_1)}$ showing all phases. Self-transitions are omitted due to readability.

Table 8.2: Values $p_{k;w}^{(v_1)}$ for $k, w \in \mathcal{Q}'$ for the Four Cross Network. Zero values greyed out.

k \ w	$\mathbf{b}_0$00000	$\mathbf{b}_1$00000	$\mathbf{b}_1$00001	$\mathbf{b}_2$00000	$\mathbf{b}_2$00010	$\mathbf{b}_3$00000	$\mathbf{b}_3$00001	$\mathbf{b}_3$00010	$\mathbf{b}_{16}$00000	$\mathbf{b}_{16}$10000	$\mathbf{b}_{17}$00000	$\mathbf{b}_{17}$00001	$\mathbf{b}_{17}$10000	$\mathbf{b}_{18}$00000	$\mathbf{b}_{18}$00010	$\mathbf{b}_{18}$10000	$\mathbf{b}_{19}$00000	$\mathbf{b}_{19}$00001	$\mathbf{b}_{19}$00010	$\mathbf{b}_{19}$10000
$\mathbf{b}_0$00000	0.9000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_1$00000	0.0000	0.4861	0.4861	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_1$00001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\mathbf{b}_2$00000	0.0000	0.0361	0.0361	0.4500	0.4500	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_2$00010	0.0000	0.0361	0.0361	0.4500	0.4500	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_3$00000	0.0000	0.0361	0.0361	0.0000	0.0000	0.3042	0.3042	0.3042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_3$00001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\mathbf{b}_3$00010	0.0000	0.0361	0.0361	0.0000	0.0000	0.3042	0.3042	0.3042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_{16}$00000	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.4500	0.4500	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_{16}$10000	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.4500	0.4500	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_{17}$00000	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.3042	0.3042	0.3042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_{17}$00001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\mathbf{b}_{17}$10000	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.3042	0.3042	0.3042	0.0000	0.0000	0.0000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_{18}$00000	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.3000	0.3000	0.3000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_{18}$00010	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.3000	0.3000	0.3000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_{18}$10000	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.3000	0.3000	0.3000	0.0006	0.0006	0.0006	0.0006
$\mathbf{b}_{19}$00000	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.2256	0.2256	0.2256	0.2256
$\mathbf{b}_{19}$00001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\mathbf{b}_{19}$00010	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.2256	0.2256	0.2256	0.2256
$\mathbf{b}_{19}$10000	0.0000	0.0361	0.0361	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0042	0.0042	0.0042	0.0000	0.0000	0.0000	0.2256	0.2256	0.2256	0.2256

Table 8.3: Values $p_k^{(v_1)}$ and $q_k^{(v_1)}$ for $k \in \mathcal{Q}'$ for the Four Cross Network. Zero values greyed out.

k	$\mathbf{b}_0$00000	$\mathbf{b}_1$00000	$\mathbf{b}_1$00001	$\mathbf{b}_2$00000	$\mathbf{b}_2$00010	$\mathbf{b}_3$00000	$\mathbf{b}_3$00001	$\mathbf{b}_3$00010	$\mathbf{b}_{16}$00000	$\mathbf{b}_{16}$10000	$\mathbf{b}_{17}$00000	$\mathbf{b}_{17}$00001	$\mathbf{b}_{17}$10000	$\mathbf{b}_{18}$00000	$\mathbf{b}_{18}$00010	$\mathbf{b}_{18}$10000	$\mathbf{b}_{19}$00000	$\mathbf{b}_{19}$00001	$\mathbf{b}_{19}$00010	$\mathbf{b}_{19}$10000
$p_k^{(v_1)}$	0.0000	0.3613	0.3613	0.0000	0.0000	0.0425	0.0425	0.0425	0.0000	0.0000	0.0425	0.0425	0.0425	0.0000	0.0000	0.0000	0.0056	0.0056	0.0056	0.0056
$q_k^{(v_1)}$	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000

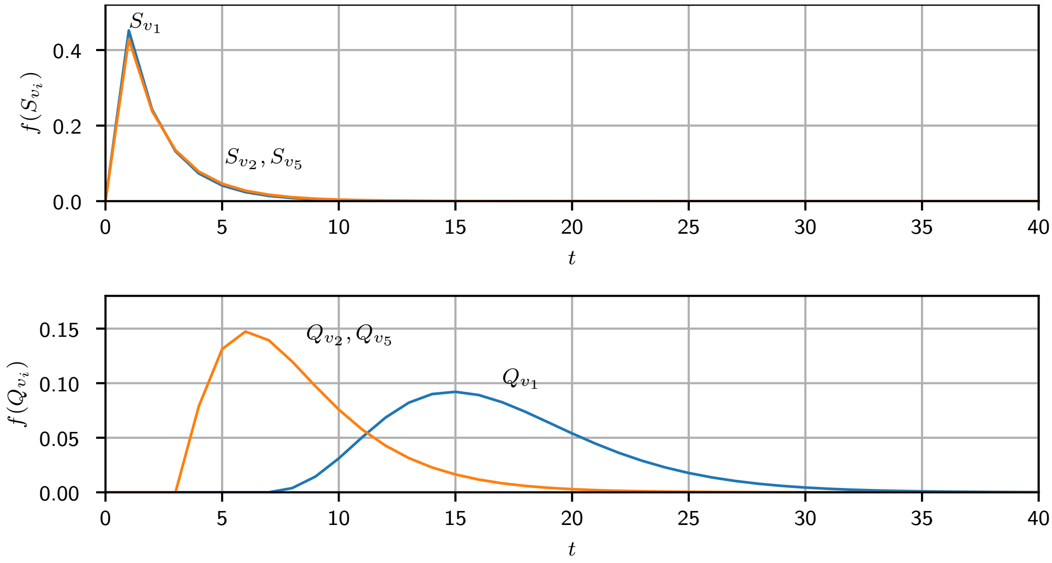


Figure 8.11: PDF of the service times S_{v_i} and queueing delays Q_{v_i}.

Tables 8.2 and 8.3 show the resulting numerical values of the transition matrix $\mathbf{P}^{(v_1)}$ for node v_1 and the initialization and exit vectors, respectively. Zero-valued entries are greyed out. The node v_1 is interesting here because it is the coordination point of the two flows. An observation that can be made is that the subnetwork $\mathbf{b}_i$ is rarely changed, which was intended with the parameter $\tau = 0.9$. Also, the probabilities for entering a phase associated with a given subnetwork $\mathbf{b}_i$ are equal for each phase associated with that subnetwork. This is he intended behavior for TDMA networks as described in Section 8.3.4. The column sums of $\mathbf{P}^{(v)}$ sum up to 1 iff $\mathbf{p}^{(v)}$ is 0 for the particular phase, or are zero otherwise.

8.5.2 Expected Queueing Behavior in the Scenario

It can be seen clearly that the node v_1 in the center of the Four Cross Network is the bottleneck in this scenario. As it conflicts with all other nodes around it, especially including the two other sending nods v_2 and v_5. The channel utilization at this node should thus be equal to (or above) the sum of the relative loads of all three nodes, as they all have to compete equally for medium access. The sum $x_1 + x_2 + x_5 = 0.6$ means that the medium must at least be utilized 60 %, as far as the channel environment at node v_1 is concerned.

Due to the simple scenario, we can assume that the service times should behave similarly to a Poisson model, although we do not know the distribution parameters. All three active nodes are within the detection range of each other, so they can coordinate themselves. More specifically, v_1 will act as the coordinator, as all single-hop transmissions involve v_1 either as a transmitter or as a receiver. The overall utilization at v_1 is a significant amount, such that we can expect the node's queues to be filled at every point in time with at least one packet. The transmission processes, therefore, will neither run dry at any point in time nor will it overload. So, it will eventually reach a steady state. Since the merging of two or more Poisson processes is again are Poisson processes, the overall system must behave like a Poisson process.

8.5.3 Resulting Service Time and Queueing Delay Distributions

From Algorithm 3 we can now determine the service time and queueing delay distributions using the expected value method. The resulting plots are shown in Figure 8.11 for the active nodes v_1, v_2, v_5. It is first observable that the ME model produces equal results for v_2 and v_5, as they are absolutely symmetric and there is no difference between the two nodes, so there is no reason that the model should distinguish between the two.

The service time S_{v_1} is similar but not equal to the other two S_{v_2}, S_{v_5}. This is clear since all medium access is coordinated with v_1, and all three nodes are similarly utilized. The expected queue length for the nodes are $\mathrm{E}[N_{v_1}] = 7.20, \mathrm{E}[N_{v_2}] = \mathrm{E}[N_{v_5}] = 3.80$. The distributions of $S_{v_1}, S_{v_2}, S_{v_5}$ are similar to a Geometric distribution, which is expected for a discrete Poisson process.

The queueing delay distributions shown in Figure 8.11 follow the expected queue length method and take the form of discrete Erlang distributions.

8.6 Discussion of the Approach

The ME model is capable of providing a means for estimating the probability distribution for queueing delay in WMHN, as we have elaborated as an essential part for modeling overall end-to-end delays in Chapter 6. It is capable of modeling the channel utilization in multi-hop networks with both managed TDMA protocols and dynamic CSMA protocols. This property has been shown by Stojanova et al. [100], from whom we have initially taken the Divide and Conquer modeling approach and integrated it into an ME queueing model for latency modeling. As the state-of-the-art medium latency models are entirely based on *M / M / 1* and the Erlang or Gamma distributions, they rely on a proper estimation of the distribution parameters. The ME model that we have derived drastically reduces the number of parameters, as the utilization is inferred by the model itself from the flow parameters and the topology and conflict graph of the modeled network.

The computational complexity of Algorithm 3, however, is quadratic both in the number of valid phases o' and the intended number of samples n_{slots} drawn from the PDF. The number of phases itself is exponentially related to the number of nodes. This relation makes the ME model computationally expensive for even moderately sized ANs. As the steady-state solution for *ME / ME / 1* queueing models is generally not expressible in analytical form [98], there is also no alternative to a numerical solution.

On the other hand, the ME queueing model represents various effects that appear on the MAC Layer in WMHN. For example, it can model and, therefore, quantify the effect of spatial reuse of CSMA-based networks such as WiFi. In addition, no assumptions on model parameters have to be made, which is a significant improvement to the state-of-the-art model based on the Gamma distribution, where a huge set of distribution parameters must be found and justified even for small-scale multi-hop networks. The ME queueing model only relies on the flow parameters of the registered haptic applications and a proper assessment of the channel capacities at the nodes, which is an easier estimation task. The set of model parameters is, therefore, much smaller compared to that of Gamma models, which, in addition, can also not model the complex interplay of the channel access of many nodes on a shared medium.

It must be shown in the future if such a model is applicable for real-world modeling problems. Arguably, it has benefits in terms of accuracy as long as the conflict graph is dense and medium utilization is above a certain threshold, which is often the case for WiFi mesh networks, for example. But if the utilization can be held low, or interference can be mitigated, e.g., with beamforming, the ME queueing model can be replaced by much simpler approaches, as we have done already in Chapter 7. In such cases, the well-known Gamma distribution is a better modeling option.

CHAPTER 9

Protocol Enhancements for WiFi: Towards Tactile Wireless Multi-Hop Networks

In this chapter, we modify the IEEE 802.11 protocol to support Haptic Communication in WiFi networks. The basis for the modification are the QoS extensions introduced by the amendment IEEE 802.11e [119], called Enhanced Distributed Channel Access (EDCA). EDCA works with a set of coordination functions (DCF, PCF, and HCF), which we extend by a specific *Tactile Coordination Function*. The TCF solely leverages existing mechanisms, which makes the extension backward compatible with current IEEE 802.11 networks. WiFi nodes without a TCF implementation can be part of a WiFi network without denying Haptic Communication for TCF-enabled nodes. We introduce a new Access Category for prioritization of haptic flows, which can be either single-hop infrastructure networks or mesh networks (WMHN). We first introduce EDCA in the first section before we detail the individual parameters used for prioritization and discuss their adaptation for Haptic Communication. We evaluate our approach in a simulation experiment.

9.1 The EDCA Mechanism

We first shortly review the basic operation principle of IEEE 802.11 [39]. WiFi, as of 2022 [39], can operate with one of three different mechanisms for channel access, which are all based on CSMA/CA as described in Section 7.2. These are the Distributed Coordination Function (DCF), the Hybrid Coordination Function (HCF), and the Mesh Coordination Function (MCF). While the DCF is basically plain CSMA/CA, HCF and MCF are adaptations towards QoS support and mesh networking, respectively. Both the HCF and MCF use an EDCA that supports QoS in a DiffServ-like environment that allows for flow prioritization. EDCA was introduced with IEEE 802.11e to support audio and video streams with enhanced QoS support and has replaced the former Point Coordination Function (PCF). EDCA is more widely adopted as it is a comparatively small extension to the original DCF. It is therefore a natural basis for an extension to provide Haptic Communication support in WiFi.

The EDCA mechanism defines four Access Categories (ACs) that provide support for the delivery of traffic with User Prioritys (UPs) at the stations. Outgoing traffic can be labeled according to one of the four ACs in order to be sorted into one of four separate output queues from which the next frame to transmit is drawn. The four queues are labeled Voice (VO), Video (VI), Best Effort (BE), Background (BK), respectively. They are assigned different priorities in that order, from highest to lowest. They are also handed to different functions, called Enhanced Distributed Channel Access Functions (EDCAFs), that determine the channel access parameters for the MAC-Layer. The EDCAF parameters control the prioritization on the channel during the contention with other nodes. Figure 9.1 depicts the setup of these queues. Each labeled frame coming from an upper layer is put into the respective queue matching the assigned label. If no label was assigned to a frame, a

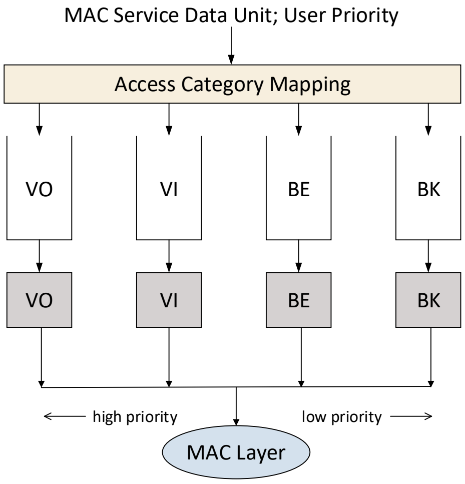


Figure 9.1: Overview of the unmodified EDCA queuing model.

default queue is used, which is normally BE. The MAC Layer chooses the next frame from a queue according to a scheduling mechanism that respects the priorities of the queues. MAC parameters are then assigned to the frame according to the assigned EDCAF.

9.2 EDCA Parameters and their Adaptation to Haptic Communication

In the long term, introducing a new Access Category into IEEE 802.11 is necessary as the interest in a haptic modality will finally also reach WiFi users. We call the new Access Category Tactile Control (TC) and include it into the set of existing Access Categories and the dedicated transmit queues. Figure 9.2 shows the mapping from the respective MAC Service Data Unit (MSDU) and UP to the transmit queues and the five independent EDCAFs.

The channel access parameters as controlled by the EDCAF can be summarized as a 6-tuple π_{EDCA} ={IFS, TXOP, CWmin, CWmax, RetryLimit, QueueSize}. The Access Categories only differ in these parameters from one another; their handling of frames is otherwise identical. In the following, we discuss these parameters in detail and introduce the adjustments that we made for the TC Access Category. Table 9.1 summarizes the results of this discussion and shows all settings we propose as the TCF supplement. As a measure of backward compatibility, we did not adjust any parameter outside of the TC Access Category, so the behavior of TCF remains the same as for unmodified EDCA for any traffic that does not fall into the haptic modality.

The settings directly impact the delay of each transmission. The single-hop delay $T_{v;u}$ of a frame between nodes v, u can be described as a function depending on the nodes, the set of EDCA parameters π_{EDCA} ={IFS, TXOP, CW, RetryLimit, QueueSize} and a noise η:

$$T_{v;u} = f\left(v, u, \pi_{\text{EDCA}}\right) + \eta.$$

Table 9.1: The EDCA settings for the Tactile Coordination Function. Haptic data uses the AC_TC category, highlighted in bold font. All other settings remain the same as in IEEE 802.11 [39].

User Priority	Access Category	AIFSN	CWmin	CWmax	RetryLimit	TXOPLimit	QueueSize
1, 2	AC_BK	7 slots	15	1023	7	0	14
0, 3	AC_BE	3 slots	15	1023	7	0	14
4, 5	AC_VI	2 slots	7	15	7	3.008 ms	14
6, 7	AC_VO	2 slots	3	7	7	1.504 ms	14
8	**AC_TC**	**1 slot**	**3**	**7**	**7**	**1.000 ms**	**14**

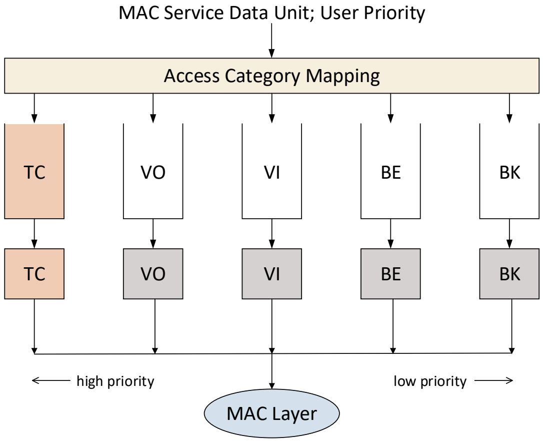


Figure 9.2: Overview of the adapted EDCA model. We introduce a new Access Category, Tactile Control (TC), with the highest priority.

The noise term η represents all non-deterministic effects.

9.2.1 Inter-Frame Space (IFS)

The IFS controls the prioritization of frames. There are several pre-defined IFSs in the standard, e.g., Arbitration Inter-Frame Space (AIFS) and Short Inter-Frame Space (SIFS). The AIFS is used by QoS nodes to initialize the arbitration phase. The shorter the AIFS, the higher the priority. A short AIFS results in a faster count down of the backoff counter and, thus, increases the priority. The AIFS is an adjustable time span which is given by:

$$\text{AIFS} = \text{SIFS} + \text{AIFSN} \times \text{SlotTime}.$$

The durations of SIFS and SlotTime are constants and defined in the standard [39]. For example, with 802.11n at 2.4 GHz, SIFS = 10 µs and SlotTime = 20 µs. AIFSN is a positive integer defining the number of slots to wait beyond the duration of one SIFS, which is determined by the Access Category. The values of AIFSN per Access Category are given in Table 9.1. AIFSN is the primary means for prioritization. Voice and video queues have smaller values for AIFSN than best-effort data to increase their priorities. Therefore, the tactile data should receive a smaller value in order to enforce the highest priority.

9.2.2 Contention Window (CWmin, CWmax)

The contention window is responsible for collision avoidance. Nodes generate an additional random backoff time if the backoff counter contains a nonzero value. The backoff time is measured in slot times (SlotTime) and drawn from a uniform random distribution within the boundaries [1, CW]. The value CW is initially set to a constant CWmin for each initial transmission attempt of a frame. CW is increased for each retransmission that is necessary. Usually, the CW value is doubled with each attempt, but it is capped at a constant maximum CWmax. When a frame is finally transmitted successfully, the backoff counter CW is reset to zero.

A small CWmin results in smaller backoff times, which makes the node access the medium more frequently. In the original EDCA settings, the voice queue is assigned the smallest range between CWmin and CWmax. However, reducing these values has both advantages and disadvantages. A small value of CWmax is desired to reduce the latency, but in the contention process, the probability of collision increases. For example, the voice queue (AC_VO) has values CWmin = 3 and CWmax = 7. CW starts with 3 for the first transmission and is doubled for each retransmission until it reaches CWmax. Thus, for the voice queue, CW is only increased two times at maximum: from 3 to 6 for the first retransmission and then from 6 to 7 =CWmax for the second retransmission. As the slot time is typically a small value in the range of microseconds, the contention window for voice data is overall short, leaving no room for further prioritization. We thus recommend a range of CWmin = 3 and CWmax = 7 for haptic frames, which is the same as for the VO category.

9.2.3 Retransmission Limitation (RetryLimit)

The retransmission limitation defines how often a frame is retransmitted in case of transmission errors. Retransmissions cause latency not only for the frame that is currently transmitted, but also for all following frames in the same queue, so a limit is necessary for latency reduction. The standard handles long and short frames differently, granting a significantly smaller limit to long frames. Since haptic frames have limited size, only the short retry limit applies to them.

To adjust this value, a rough estimation of its impact is helpful. For a frame of 1000 bit in length, a data rate of 600 Mbit/s would allow for 600 transmission attempts in one millisecond. A heavy reduction of the retransmission limit for TC, therefore, can not be assessed to have a significant impact on the overall latency. On the other hand, in our experience, the standard limit of 7 already results in a remarkably low loss rate [120, 121], so we also do not increase the limit.

9.2.4 Transmission Opportunity Limitation (TXOP)

The TXOP limit allocates channel resources for a node for consecutive transmissions. When a node receives a transmission opportunity, it is granted a time window in which it can transmit several frames at once to either the same or to different destinations. The overhead of medium arbitration is therefore avoided, resulting in an overall better medium utilization. The TXOP value specifies the length of the transmission opportunity window, and it is entered whenever a node receives access to the medium. In addition, a high TXOP value for an Access Category results in a higher priority, as more frames can be transmitted.

Due to the fixed allocation of the TXOP limit, some limitations are encountered. Tactile Internet Applications are usually bi-directional, and thus require both a control and a feedback flow, which have the same requirements and same priority. These two flows compete with each other, so a transmission opportunity from one side blocks the other side from its channel access. As Tactile Internet Applications transmit only small amounts of data, they do not require long transmission windows. We reduce the TXOP limit of TC to 1 ms to ensure a minimum frame loss and reduction of queuing delay.

9.2.5 Queue Size Limitation (QueueSize)

In general, queueing delay depends on queue size, so a limitation of the overall queue length can reduce the delay. As queues with high load can be considered as overloaded, and delayed frames for Haptic Communication can be rather dropped than transmitted with high delay, a limit can be applied. Moreover, Haptic Communication is generally isochronous, so lost frames due to overfull queues can be replaced by following data frame. The standard defines a queue length of 14 for each Access Category. The reduction of this value can lead to significant losses, and we do not consider reducing it for the TC Access Category, as we can model queueing delays with our ME model. High queueing delays should be considered as overload situations, so the reduction of queueing delays should be solved on a higher layer

Table 9.2: Overview of the used simulation parameters. Sets of values indicate variations between different runs, resulting in 20 runs total.

Parameter	Value
Simulation time	4 s
Hop distance	50 m
Transmitter power	100 mW
NIC mode	802.11n 2.4 GHz
Max. bitrate	600 Mbit/s
Bandwidth	40 MHz

Parameter	Haptic data	Background traffic
Hop count l	{ 1, 2, 3, 4, 5 }	1
Access Category	{ VO, TC }	{ - (not present), BK }
Payload size s	192 bit	8000 bit
Sending interval	1 ms	0.026 ms
Packets total	4001	154 001

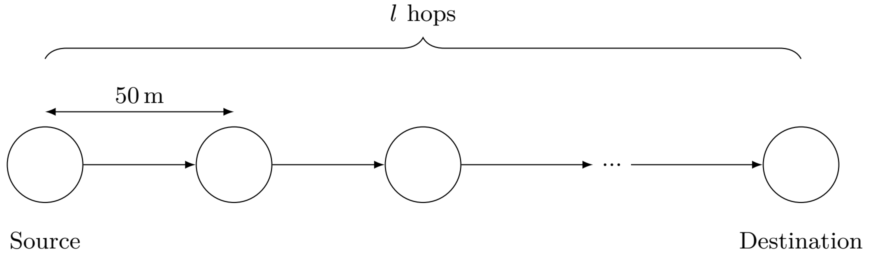


Figure 9.3: Topology of the simulated networks.

rather than by adjusting low-level transmission parameters. We thus leave the queue size at the default value of 14.

9.3 Evaluation in Simulation

We evaluate our approach in a set of simulation experiments. We use OMNeT++ as our simulation tool to simulate a multi-hop WiFi network with Haptic Communication. We analyze the latency and jitter behavior, and also evaluate the packet loss.

9.3.1 Simulation Setup

We have two distinct experiment groups. In the first group, we simulate a haptic flow without any background traffic. In the second group, we additionally introduce some background traffic. The background traffic is configured to generate approximately a 50 % channel utilization alone. We study the behavior of the haptic data in the network, which is expected to be unaffected by the onset of the background traffic. When prioritization with EDCA functions correctly, the same results should be observed in both experiment groups.

We furthermore compare the existing HCF to our new TCF approach. HCF can offer prioritization already with 8 different Access Categories, which could probably be re-purposed for haptic data. The HCF module used in our experiment is identical to the unmodified EDCA mechanism (excluding the AC_TC Access Category) as shown in Table 9.1. Generally, we are interested in multi-hop behavior, so we experiment with different network sizes. The hop count varies between one and five. All our networks have a linear topology as depicted in Figure 9.3. The simulation settings for OMNet++ are summarized in Table 9.2.

The priority allocation in all our experiments is based on the IP port. On port 5000, the BK priority is chosen for frames both with TCF and HCF. Port 5500 is reserved for haptic data, which for TCF is mapped to the TC Access Category. For HCF, port 5500 is mapped to the highest priority category available there, which is VO.

9.3.2 Simulation Results for Latency

Latency measures of HCF and TCF are shown in Figure 9.4, and detailed values in Table 9.3.

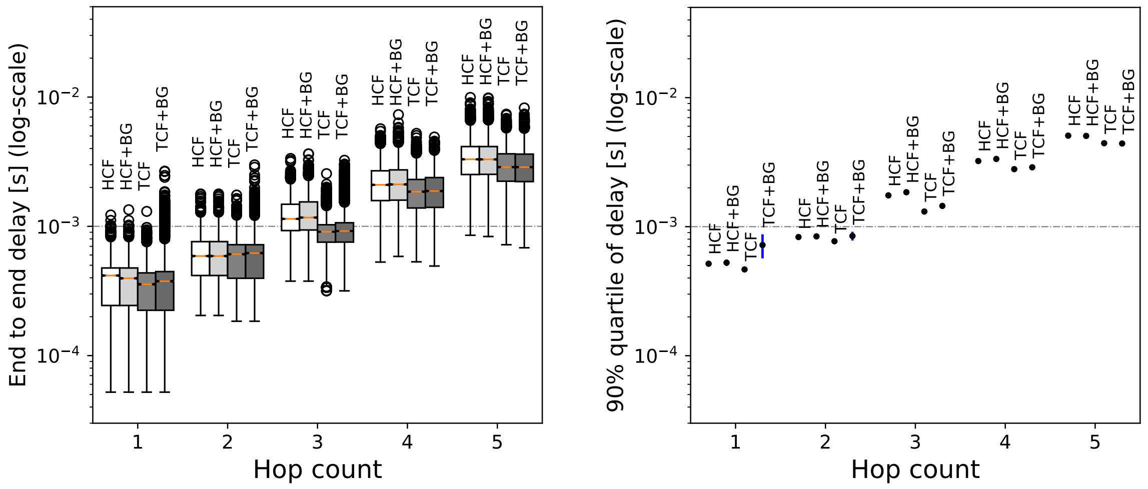


Figure 9.4: Simulation results for haptic flow delay with varying hop count (n = 4001). Shown are box plots (left) and the estimators of the 90 % quantiles with error margins (α = 0.01, right). With the Tactile Coordination Function (TCF), the mean latency is within 1 ms in the three hop network.

Table 9.3: Mean end-to-end delay of haptic flows ($n = 4001, \alpha = 0.01$). Values given in ms.

	Without background traffic					With background traffic				
CF	1 hop	2 hops	3 hops	4 hops	5 hops	1 hop	2 hops	3 hops	4 hops	5 hops
HCF	0.379	0.584	1.206	2.177	3.419	0.373	0.591	1.251	2.206	3.422
	± 0.006	± 0.009	± 0.016	± 0.032	± 0.05	± 0.006	± 0.009	± 0.018	± 0.034	± 0.051
TCF	0.346	0.577	0.918	1.890	2.976	0.407	0.624	0.972	1.938	2.967
	± 0.005	± 0.008	± 0.011	± 0.027	± 0.043	± 0.012	± 0.012	± 0.015	± 0.029	± 0.043

Figure 9.4 also shows that the mean delay of haptic data gradually increases with the hop count, as is expected. As can be observed, even the standard HCF can already support low levels of mean delay in the vicinity of below 1 ms. This is due to the already comparably short backoff times of the selected VO Access Category. With TCF, however, the mean delays could be reduced even further in all cases. The average delay with TCF is within the 1 ms range in the case of three hops, while with HCF the average delay is slightly larger than 1 ms. In general, the experiment shows the overall applicability of Haptic Communication in WMHN. The limit seems to be three hops, but although a significant latency increase is observable with four and five hops, some applications with lower requirements may also be valid here.

We also have evaluated the 90 %-quantiles from the same data sets, as shown on the right-hand plot of Figure 9.4. The corresponding data points are summarized in Table 9.4. The 90 %-quantiles provide a better understanding of the real-time capabilities of the protocols, as an application can estimate what delays can be achieved 90 % of the time. The differences between HCF and TCF show here to be significant, beginning at a hop count of three. We can conclude that TCF outperforms HCF in terms of both mean latency and 90 % quantiles. Additionally, the onset of background traffic has only a small impact on the latency, which shows that the prioritization works as intended. Although a significant difference is present between the no-background and background scenarios, the EDCA mechanism can not be expected to provide a clean separation and prioritization.

9.3.3 Simulation Results for Jitter

For measuring jitter, we use the standard deviation of the end-to-end delay. The jitter comparison is shown in Figure 9.5. The numerical results are given in Table 9.5.

Table 9.4: 90 %-quantiles of the end-to-end delays of haptic flows in the same experiment ($n = 4001, \alpha = 0.01$). Values given in ms.

	Without background traffic					With background traffic				
CF	1 hop	2 hops	3 hops	4 hops	5 hops	1 hop	2 hops	3 hops	4 hops	5 hops
HCF	0.516 ±0.020	0.831 ±0.010	1.748 ±0.057	3.225 ±0.134	5.073 ±0.209	0.526 ±0.030	0.841 ±0.020	1.848 ±0.066	3.348 ±0.145	5.049 ±0.210
TCF	0.466 ±0.010	0.771 ±0.010	1.311 ±0.066	2.791 ±0.104	4.434 ±0.181	0.720 ±0.152	0.847 ±0.066	1.448 ±0.073	2.885 ±0.102	4.410 ±0.176

Table 9.5: Mean jitter of haptic flows ($n = 4001$). Values are given in ms.

	Without background traffic					With background traffic				
CF	1 hop	2 hops	3 hops	4 hops	5 hops	1 hop	2 hops	3 hops	4 hops	5 hops
HCF	0.145	0.218	0.396	0.790	1.233	0.150	0.223	0.439	0.836	1.238
TCF	0.130	0.191	0.279	0.661	1.051	0.290	0.298	0.369	0.703	1.050

When no background traffic is present, both HCF and TCF jitter increases proportionally to the number of hops. TCF jitter is, however, significantly smaller with higher hop counts. We address this effect to the reduction of the TXOP limit, which decreases the randomness of the process. This is an indicator that this limit can be reduced even further eventually. As elaborated, Haptic Communication might not profit from the transmission opportunity concept either.

With background traffic, HCF jitter shows almost no change. However, the traffic has a significant impact on the jitter of TCF at low hop counts. With increasing hop count, the TCF jitter approaches gradually that of TCF with no background traffic. It can be said that with a high traffic load, TCF still has some advantages in terms of jitter over HCF on longer paths.

9.3.4 Simulation Results for Frame Loss

Table 9.6 summarizes the frame loss during our experiments. HCF and TCF show fewer losses (out of the $n = 4001$ messages sent in total). Although the numbers increase with active background traffic, the loss rate is far below 1 %, which is within a tolerable range for some teleoperation applications. The maximum of 4 lost messages with TCF at 5 hops yields a 0.1 % loss rate or one lost message in a second, respectively.

Table 9.6: Packet loss ($n = 4001$).

	Without background traffic					With background traffic				
CF	1 hop	2 hops	3 hops	4 hops	5 hops	1 hop	2 hops	3 hops	4 hops	5 hops
HCF	0	0	1	1	2	0	0	2	5	6
TCF	0	0	0	1	3	0	0	0	2	4

9.3.5 Worst Case Delay Analysis

In this section, we analyze worst-case delay of the TCF contention process. It is composed of three parts: the IFS, the backoff process, and the transmission process itself. Their durations are controlled by three variables: IFS, contention window sizes, and the payload size. Figure 9.6 shows a timing diagram of the relevant fragment.

In the IFS phase, the TCF delay (AIFS = 30 µs) is shorter than the HCF delay (AIFS = 50 µs) by one slot. The length of the backoff window is determined by the range of the random CW, which varies between 3 and 7. The delay in the transmission phase depends on the size of the transmitted data. In the worst case, the contention delay thus consists of an AIFS and 7

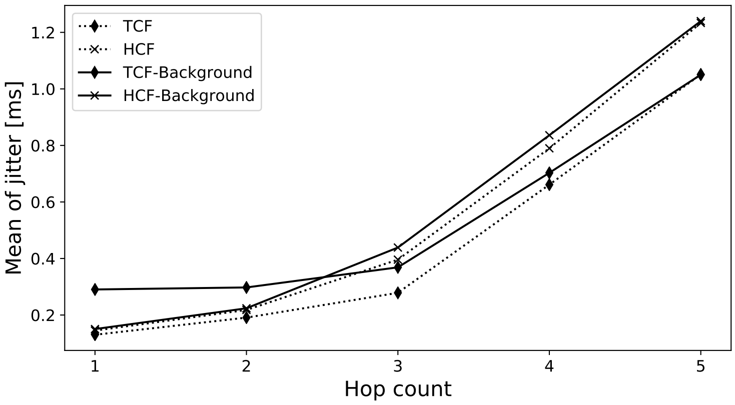


Figure 9.5: Simulation results for jitter of haptic flows in relation to the hop count.

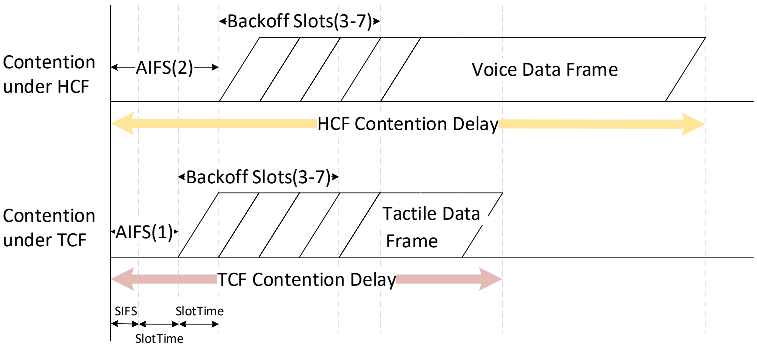


Figure 9.6: Contention process comparison of HCF and TCF.

backoff slots. The maximum backoff time is 7 slots for both the highest Access Category VO with HCF and for TC with TCF. VO and TC transmissions need a TXOP of 1.504 ms or 1 ms, respectively, as shown in Table 9.1. The TXOP lengths must be added to the calculations, as the worst-case is given when a frame is sent at the last chance of a TXOP.

Table 9.7: Comparison of worst-case contention delays.

CF	AIFS	Max. backoff	TXOP	Delay
HCF	50 µs	140 µs	1.504 ms	1.694 ms
TCF	30 µs	140 µs	1.000 ms	1.170 ms

The values for AIFS are calculated as follows:

$$\begin{aligned}\mathrm{AIFS}_{\mathrm{HCF}} &= \mathrm{SIFS} + 2\,\mathrm{Slots} = 10\,\mu\mathrm{s} + 40\,\mu\mathrm{s} = 50\,\mu\mathrm{s}\\ \mathrm{AIFS}_{\mathrm{TCF}} &= \mathrm{SIFS} + 1\,\mathrm{Slots} = 10\,\mu\mathrm{s} + 20\,\mu\mathrm{s} = 30\,\mu\mathrm{s}\end{aligned}$$

9.4 Discussion

In this chapter, we proposed the introduction of a Tactile Coordination Function (TCF) to WiFi, that supports the transmission of haptic flows in WMHN. TCF is an extension to set of existing control functions in WiFi and is backward compatible with the standard, such that nodes supporting TCF will not interfere with nodes that only support HCF and vice versa. In the transmission process, haptic data is assigned a higher priority. Haptic data thus receives shorter delays and lower jitter than with the existing protocols. We have achieved this prioritization basically by leveraging reserves that are present in the standard and adapting the EDCA mechanism to the needs of Haptic Communication.

Our approach shows that Haptic Communication is no different in any regard from other well-known data modalities, such as voice or video data, and that exisisting communication protocols can be retrofitted to support this new modality. In a simulation experiment we show that already current IEEE 802.11n MIMO technology supports the requirements of haptic flows for various applications. The 1 ms, 1 kHz requirements could be met in a 3-hop network in our experiments. Reliability requirements, though, might be a more critical problem, as we could only achieve a 99.9 % packet delivery rate. In addition, five- and four-hop flows show a substantial increase in latency and jitter. Some applications, though, with lesser requirements can operate there as well. These are promising results that justify further research on haptic flow transmission over CSMA-based WMHN.

As a result of this chapter, we assess that the prioritization of Haptic Communication with TCF as an amendment to IEEE 802.11 is feasible. WiFi thus shows as a potential competitor to mobile telecommunication systems, such as 5G, even with the strict QoS requirements given by IMT 2020. WiFi has the capabilities to be applied for the URLLC use case. In Chapter 11, we show this capability again with a testbed experiment, where we implement the proposed TCF amendment within the Linux mac80211 subsystem with real hardware. The results of this chapter can be repeated in the real-world testbed.

PART IV

Application Framework

CHAPTER 10

A Haptic Application Framework and Testbed

In this chapter, we introduce the Haptic Digital Twin Framework (HDTF) for Tactile Internet Applications in detail, which we have sketched briefly in Chapter 5. The Framework is aimed at minimizing code duplication that arises when software needs to address pairs of real and virtual objects, as is required with Digital Twins. A concrete realization in the form of source code, however, at this early stage of technology adoption is difficult. As a step towards such a realization we propose the Haptic Communication Testbed at the Otto-von-Guericke University of Magdeburg (OVGU-HC), that implements the Digital Twin interface of HDTF and can be used in conjunction with the MIoT-Lab [66] to form a complete end-to-end testbed for Tactile Internet Applications.

10.1 Motivation

With the preceding chapters, we have developed the foundations for engineering latency-aware systems within the Access Network and developed a haptic protocol for the WiFi MAC Layer. The HDTF uses these two contributions for introducing a Software architecture respecting the distinction between Access Networks and the Tactile Internet Backbone as described in Section 6.1. The central role comes to the TSEs in the Edge Cloud that are an interface between the two network roles, Access Network and Backbone. TSEs have to serve two different purposes: First, they host Digital Twins for interaction with the respective peers at the wireless edge, and second, they organize the synchronization of the Digital Twins, which appear at both sides of the application, which requires a steady information flow through the Backbone.

For both parts, application developers have the problem that they need to implement services, protocols, and drivers for entities that are either virtual or real objects. Realizing a true Digital Twin requires that both read and write access must be agnostic to the fact whether the real or the virtual part of the Digital Twin is addressed. Often both parts do not have the same API, as virtual objects are often too abstract to allow for an in-depth copy of the real API. Sometimes also the real API is overly complex due to proprietary or legacy drivers, wear-induced anomalies, or custom Hardware modifications. Some virtual objects are often just databases that save the last state of a real object, while access to the real objects requires complex interactions using sensors and actuators. While it is technically plausible to implement two different interfaces dedicated to either virtual or real entities, and to decide with a query which interface to use, it is an inefficient way that will produce many code duplications and boilerplate code. In the end, application designers would have to implement two more or less fully capable applications with this method. In addition, an application may implement not just two, but several kinds of roles, depending on the complexity of the Digital Twin.

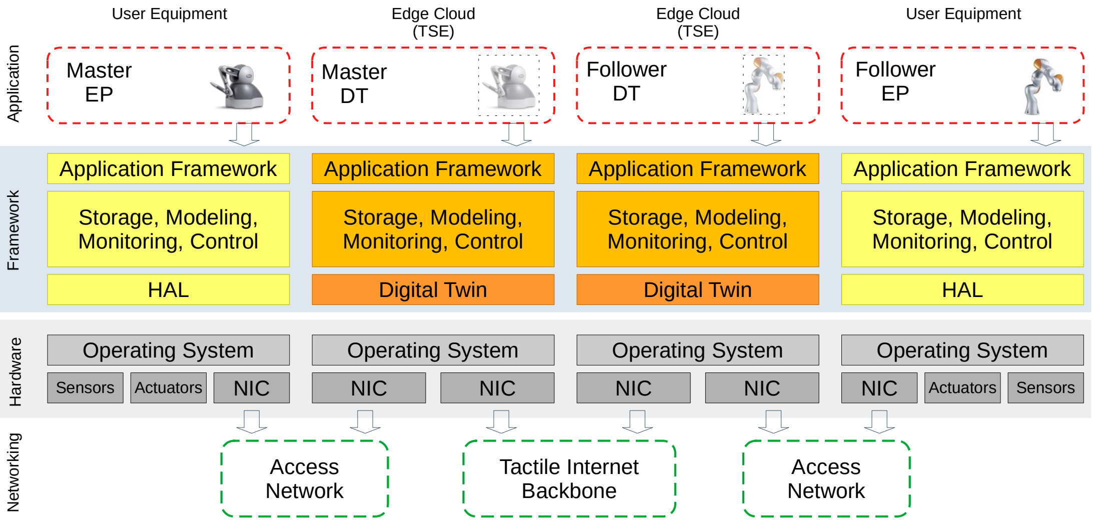


Figure 10.1: Components of the Haptic Digital Twin Framework.

The problem can be addressed as a *hardware abstraction* problem. Although the virtual part of a Digital Twin is not a piece of hardware in the classical sense, it behaves very similarly. With the HDTF, we show that virtual and physical spaces of Digital Twins can be abstracted like computing resources in Operating Systems, or physical resources like in the Robot Operating System (ROS) [122]. Such systems also have the ability to abstract Digital Twins from real hardware. For example, ROS offers the ability to operate simulated robots using tools like Gazebo [123] or CoppeliaSim [124]. The interfaces to simulated entities and real ones are equal, which provides a maximum of transparency and allows for using the same application code for real hardware and Digital Twins.

The HDTF can be regarded as a more lightweight and a more general middleware than, for example, ROS, which is dedicated to Tactile Internet Applications.

10.2 Framework Overview

As elaborated, Tactile Internet Applications consist at least of four parts, in contrast to traditional Internet applications that mostly have a client/server distinction. As described in Chapter 6, traditional clients are replaced by master and follower endpoints (EPs). Digital Twins for master and follower, on the other, replace the role of traditional servers.

Like all communication platforms, any Tactile Internet platform can be represented in a layered fashion with communication roles at the bottom and application roles at the top. Such a layered approach is depicted in Figure 10.1. In between the communication and application roles, hardware and software components are organized as a middleware to provide a PaaS solution for the application roles. The seamless integration of hardware and software is essential for the Tactile Internet, as the 1 ms-bound requires applications to migrate quickly in order to reduce physical distances. The middleware, therefore, must allow the applications to operate in a location-agnostic way, which means that their physical location (in terms of IP address) must be able to change with little configuration overhead, and potentially even during runtime. This is a typical problem in the field of (edge) cloud computing.

New issues must, however, be addressed with the network roles at the bottom of the layered architecture. In (edge) cloud platforms, the goal is to abstract away the network entirely, which is not possible for the Tactile Internet. The role of the Tactile Internet Backbone is different from the Access Layer, which introduces two different kinds of communication – an issue that is unknown within cloud computing approaches. As we have shown within the main part of this thesis, especially the Access Network role is complex and has a strong influence on the upper layers, such that the behavior of this component must be taken into account in the middleware. The 1 ms-problem is solved mainly within the Access Network role, so the problem of finding suitable locations for the DT application roles is one of the major issues of the middleware. Quality of Service and Quality of Experience metrics have to be monitored constantly in order to support the function of the applications. And, additionally, the Digital Twins have to synchronize constantly, which requires a certain infrastructure from the middleware.

In the following, we investigate the aspects of the individual roles and parts of the infrastructure.

10.2.1 Application Roles

The four different application roles have been introduced in the formalization in Section 6.1. They differ between endpoints and Digital Twins.

Endpoints are the “user interface” of an application, running the specific application logic, and offering user or machine interaction. Typical components are control loops that control local systems, such as robotics hardware, displays, and input devices. When an endpoint is connected to the system, it maintains both a control data flow to the remote endpoint to which it is connected, but otherwise mainly communicates to the Digital Twin of its communication peer to which it establishes an URLLC connection. The implication is that the hard core of the application logic is concerned about the URLLC communication aspect. Endpoint logic must rely on a stable and reliable communication service, which is agnostic to migration operation of the Digital Twin.

Assuming that a wireless endpoint is mobile, and its position at the initialization phase of the application is unknown, a TSE within its Access Network must be found that can fulfill the QoS requirements of the intended application and has free computational capacity. The fulfillment must be checked with a model such as we have introduced in Part III. On the other hand, applications must also be aware of the influences of the network on the Quality of Experience. The network may not be able to guarantee hard 1 ms service guarantees, especially in Smart Home environments, or with potential wireless multi-hop connections in the Access Network Plane – a scenario that can happen with both WiFi and 5G connections. The QoS models developed in this thesis can be used for this assessment, and specific actions can be taken by applications in case of violations.

Conceptual consequences for the middleware are manyfold. It must supply more connectivity information to the endpoint role than in traditional cloud computing environments, where such information is mostly not provided at all, or in limited amounts in case of media streaming protocols. Such information must be provided out of band, and constant monitoring and assessment is necessary. In addition, connectivity to the Digital Twin role must be established, secured, and managed by the middleware itself, as no location information can be given. In this case, also means of replication should be included in order to rapidly switch between service providers in case of an error.

The Digital Twin role is not just a direct virtual copy of the respective communication peer, but must also include control elements that close the control loop between the local endpoint and the Digital Twin. Therefore, the Digital Twin role is a piece of software that must be provided by application developers and can not be automated by the platform. The Digital Twin itself can be provided by the middleware, which is shown as a separate layer that replaces the Hardware Abstraction Layer (HAL) for the TSE part of the framework. However, the actual modeling and part aggregation must also be provided by the application programmer. Software tasks that have to be provided by the Digital Twin role can include 3D modeling and model updates, semantic descriptions (Ontologies), state updates, state synchronization, caching (e.g., dictionaries of tactile surface information), and others. The purpose of these services is mainly the prediction of the response of a remote endpoint to stimuli induces by the local endpoint, which commonly involves complex computations and methods of machine learning and artificial intelligence. As the concrete measures are application-specific, the middleware must offer an appropriate toolset that allows application designers to quickly implement the necessary application parts.

For the Digital Twin role, the concrete requirements and software structure can hardly be specified in necessary detail at the current state of technology and research. The growing interest in the topic and the upcoming evolution in the market of Internet platforms will manifest more concrete tools and software libraries over time. We expect new approaches emerging first as closed platforms dedicated to small niche applications, setting the cornerstones for bigger, more generalized platforms later. Modern web-related paradigms, however, such as microservice architectures, serverless programming, and event-driven server logic will likely be integrated from the beginning, allowing for scalable deployment.

10.2.2 Network Roles

The network roles at the bottom of Figure 10.1 are not directly under the control of application developers, as the network is mostly provided by ISPs. However, two aspects are important for application developers. First, the two roles differ in their purpose and have effects that affect the applications, which makes it necessary to open up their structure and feed back network information into the applications. Second, the layers of the framework should be thin to enable small-footprint protocols to be developed. In conclusion, the framework should be transparent from top (application roles) to bottom (network roles). The use of open protocols on all layers enables low-overhead and homogeneous services between network layers and entities, which is necessary to assess the Quality of Service through the measures introduced in this thesis. Ultimately, open protocols allow for clear interfaces across all layers. Since network effects must be reduced to a minimum, the openness of the network is a major aspect of the future Tactile Internet.

In some cases, developers may also wish to influence the network. For example, the reservation of network resources may be critical to ensure guaranteed service qualities. Such an influence is often not possible with cloud-based infrastructure, where the main goal is to abstract application logic from the network logic. However, service guarantees and resource reservation might be a key to enabling Tactile Internet Applications. Another aspect can be the virtualization of network functions or topologies, which can be used to outsource certain application logic into the network, or to employ a custom networking regime. The virtualization part often becomes an important aspect in largely distributed applications that need service guarantees or a custom topology.

Since the transparency and customization of network functions is essential to the Tactile Internet, the middleware must allow for the easy and transparent exchange of network protocols, and individual network configuration. It must be completely protocol-agnostic, in the sense that it should enable to configure individual socket parameters, or even allow for customization of the socket creation process itself. POSIX sockets have proven to be a portable and clean interface for computer networks, allowing for abstraction of the underlying network interface, VPN or tunneling settings, but still allow for adjusting protocol settings. The middleware should offer a thin and clean interface around those sockets to allow for adaptation to the individual network roles.

10.3 Framework Components

The middleware must offer various services that can be categorized into hardware abstraction, application, and storage/modeling/control, as shown in Figure 10.1

10.3.1 Hardware Abstraction

Hardware abstraction is a necessary lower-layer component that manages access to the underlying operating system and its device drivers and system components. It is a common task that also occurs in a similar form in cloud infrastructures. But in the HDTF, the hardware abstraction also includes network models, sensors, actuators, and Digital Twins, and thus must offer a broader range of functionality. Both the endpoint and the Digital Twin require hardware abstraction. On the endpoint, sensors and actuators must be accessed, while on the Digital Twin, the same components are purely virtual. An ontology describes the linkage between the two that allows for correct addressing of all hardware resources. Drivers need to be integrated such that the software interface for virtual and physical resources are the same and produce no side effects or artifacts.

The hardware abstraction includes network resources, too, since the QoS models for the network roles need to be evaluated in a protocol-agnostic manner. Communication resources include flow definitions, quality metrics, and service descriptions. Service discovery is also necessary since the application roles need to be location-agnostic. Additionally, the HAL must decide about a handover to another TSE, which requires network management logic.

On the Digital Twins, the HAL is a purely virtual interface that replaces the respective functions of the sensors and actuators of the peer endpoint. Here the HAL does not include device drivers, but consists of interfaces to a simulation of the real entity of interest, replacing real sensors and actuators with a simulation of the physical interactions. Respective simulation infrastructure can be included and offered by the middleware. The infrastructure should allow for automatic creation, replication, deployment, and migration of these virtual environments, as soon as services are instantiated, migrated, or stopped.

10.3.2 Storage, Modeling, and Control

The storage, modeling, and control infrastructure manages system resources and their horizontal integration. Network flows and application resources from the server infrastructure have to be linked across system boundaries to create a managed application entity. The resources can include storage in form of databases, caches, or other memory, computational resources, models such as Ontologies or simulation models, and DevOps resources. The goal of this framework component is the deployment of applications, as well as their migration within servers in the edge cloud (TSEs), devices, and servers.

Configurations must be described through description languages to migrate and deploy applications. Virtualization methods are required to encapsulate application roles into sandboxes and provide differentiation between services and assign resources to applications.

10.3.3 Application Framework

The application framework located at the top level of the middleware offers interfaces for the implementation of the application logic. It enables access to the platform infrastructures of the lower layers through an abstraction of the control models.

One important primitive of this layer is the control loop, which can have its ends on both sides of the EP/DT relation. Both EP and DT can act as the controller or the control plant of a control loop. The interface of a control loop requires a data model definition, a state definition, a control algorithm, and a mapping to the respective actuators and sensors. The data model acts as a message format that is exchanged between the plant and the controller. It can be composed of primitive data types, such as floating-point values, 2D/3D image data, or others. Control loops can be open or closed. Open loops occur mostly with tactile data, while closed loops are related to kinesthetic data. The exchanged data in both cases is time-series data, that is transmitted isochronously. Pre-defined primitives can utilize haptic compression schemes to reduce the communication overhead.

Additional primitives are defined by the storage, modeling, and control services in the management infrastructure. These services are similar to those found on current cloud computing platforms. They can include database access (relational databases, table storage, or time-series databases), event logic, message queues, worker services, and others.

10.4 The OVGU-HC Haptic Communication Testbed

The HDTF, due to the novelty of the Digital Twin aspect and its early stage of technology, can not be realized or specified in detail at the time. However, we want to address the implementation of a framework for a more specific use case in the rest of this chapter.

We have now related the novel structure of Tactile Internet Applications to state-of-the-art cloud computing infrastructures, and to this extent, we have mapped most aspects of these applications to well-known and developed fundamentals. However, the aspect of the interaction between the Digital Twins and the endpoints requires still more investigation. Software components, such as control algorithms, haptic coding, and signal processing often need to be developed for both of these target platforms similarly, with only minor changes. Therefore the framework must allow for code reuse by having a clear distinction between application logic and the code related to the individual role (DT or EP).

To demonstrate such a distinction in the form of a concrete framework proposal, we chose to develop a testbed for Haptic Communication. Such a testbed must provide solutions to the same problems as stated before: It must allow for quick development of Haptic Applications and thus must separate the system logic (experiment deployment, data collection, evaluation, etc.) from the application logic itself. In addition, use cases for this testbed can be either real-world experiments conducted on a hardware platform (e.g., a robotic arm), or a simulation, which requires an abstraction between the physical and the virtual world.

The developed OVGU-HC testbed allows for repeatable evaluation of Haptic Applications as part of the MIoT-Lab, introduced in Section 3.4. It is an experimentation platform that is specifically designed around the typical data flows and communication needs of Haptic Applications. It offers an application framework for fast experiment design that is based on the networked control loop as a fundamental application primitive. In this section, we describe the OVGU-HC Framework (OVGU-HCF), we show an example experiment to demonstrate a case study, and we show a draft IEEE 1918.1 protocol definition within the OVGU-HCF.

10.4.1 Testbed Requirements and Purpose

We first give an overview of the design goals and requirements for the testbed. Haptic Applications require strict QoS provision from an underlying network, such that their performance is closely coupled with the network performance. To validate an application for its applicability and overall functionality, both the network and the application logic have to be tested together. In the classic case, without a testbed where both roles are integrated, the validation process is limited. The only feasible method would be to first translate the constraints of a control application into required communication flow parameters. Then, the data flow can be replicated and simulated within a communication testbed. This approach has its limitations due to the many involved simplifications. Considering, for example, the CPS depicted in Figure 10.2, a simple question like “how does the network affect the precision of the robot?” is nearly impossible to answer by just analyzing or simulating single components on their own. With a testbed that both covers the network and application domains, however, this question can be answered, as a full picture of the application is available. Such an “end-to-end” Haptic Communication testbed allows for assessing and measuring the influence of network effects on the performance of the application domain. It allows to measure effects of packet losses, delayed packets, and interference on the application on a higher level, and can then assess overall system safety and proper function. The development of Haptic Applications should, thus, go hand in hand with experimentation on testbeds.

Testbeds also address the problem of reproducibility of real-world measurements [125,126]. Engineers can experiment repeatably and consistent on a wide set of network topologies and communication technologies. Testbeds can also provide guidance for experiment design, including definition, execution, data collection, and evaluation. For this, they need to be general enough to not impose any restrictions on application design. For haptic CPSs, there is currently a shortcoming regarding this flexibility. One major missing feature is the missing

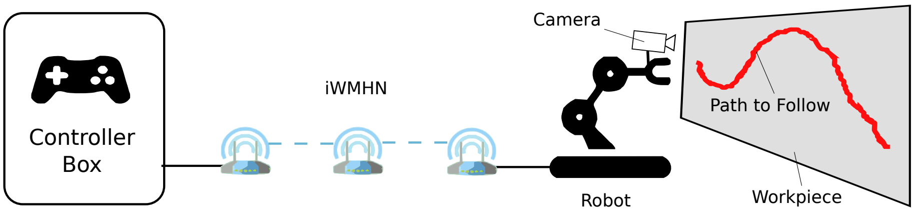


Figure 10.2: Application scenario: A line following robot as part of a CPS. The robot is controlled remotely utilizing industrial wireless multi-hop network (WMHN) infrastructure.

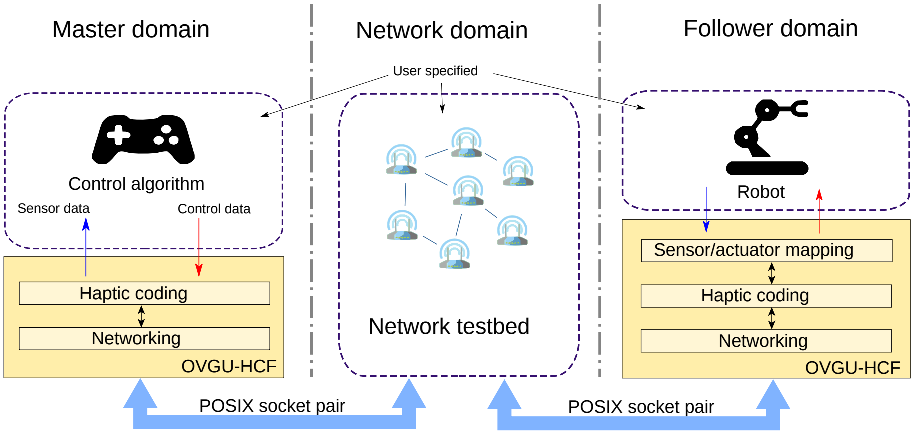


Figure 10.3: Structure of a haptic experiment. The applications in the three domains are user-specified, while the OVGU-HCF acts as a thin layer for providing portability.

support for haptic coding in testbeds so far, as this topic touches both the robotics and the network domain.

The advances in the field of Haptic Communication over the last years show that the assessment of solutions depends on the availability of real-world data for experimentation [6]. Codecs, such as the perceptual deadband filter, strongly depend on the specific application, and proper parameter adjustment is difficult. The OVGU-HC intends to address the aforementioned issues by covering the following aspects relevant for developers and experimenters:

a) automation of experiment scheduling and deployment using an experiment description language,

b) automation of collection of results and support for evaluation,

c) specification of control and sensor data flows, and

d) mapping of sensors and actuators to the control data flows.

The OVGU-HC is a part of the MIoT-Lab [66,67], introduced in Section 3.4. The MIoT-Lab already covers the points a) and b) as it automates the process of experiment node allocation, network setup, and software deployment. This is a time-consuming task normally, but with in the defined experimentation process it can orchestrate hundreds of network nodes simultaneously. The MIoT-Lab is built around a TBMS that delivers the necessary infrastructure and services for managing and controlling nodes and experiments.

The remaining points c) and d) from the above list are specific to haptic experimentation and are solved by OVGU-HC. Haptic experiments are characterized by their characteristic traffic shape, and the control loop as their core communication primitive. Although the flow structure in OVGU-HC has no restriction, it forces a clear definition of the data flow around the control loops. Each message is defined by an individual data structure which is meant to represent a changed state variable on either of the two ends.

10.4.2 OVGU-HC Components and Structure

The OVGU-HC covers three domains as depicted in Figure 10.3). The structure is borrowed from the HDTF overview in Chapter 5, indicating that the OVGU-HC is an implementation of that structure. Similar to the definition in Chapter 5, the *follower domain* hosts the teleoperators to operate with the physical world. These can be general systems for human-to-machine or machine-to-machine interaction. Often it is a robot or a CPS. The *master domain* hosts the operator controlling the follower, who can be either a human or a machine as well. In between the two is the *network domain* which represents the communication to exchange data between the two endpoint domains. The network of the testbed can consist of different types of systems, for example, an Internet connection, an IEEE 802.11 WMHN, a Bluetooth network, or a wired connection such as a CAN bus. Also, a 5G campus network is part of the OVGU-HC network domain.

The OVGU-HCF framework guides the application design and enables applications development in C/C++. It is minimal layer needed for ensuring the portability of applications between the various devices used in the MIoT-Lab. Additionally, it defines the data flows between the endpoints. From the OVGU-HC, executable binaries can directly be compiled and then be deployed by the TBMS to the various testbed nodes. We describe the OVGU-HCF in the following in more detail. The development is not yet completed, but it is intended to be published under an open source license. The following description is, therefore, not covering any implementation details, but gives a more high-level description of its components. In addition, we refer the reader to the respective literature for further information on the TBMS [125] and on the DES-Cript [127] experiment description language, which we also do not cover here.

10.4.3 Application Use Cases

Figure 10.4 shows some of the prominent components from in the three domains that can be combined individually to form a single experiment. Currently, the testbed supports haptic teleoperation, CPS in industry or smart factory scenarios, telepresence, and cooperative control, among others. We strive to cover as many use cases as possible, and more will be integrated in the future. The application logic as part of an experiment is always completely user-specified and is thus not bound to any pre-defined scenario. To setup a specific experiment, the testbed user has to implement the user-specified endpoint applications, configure the data format of the exchanged messages, and settle some parameters for the OVGU-HCF. The experiment is then compiled into executables that are deployed by the TBMS.

10.4.4 Data Model

We have adopted the data model from Steinbach et al. [6], that we have already introduced in Section 2.2. Both sides basically exchange vectors of position, velocity, or force values as part of a control loop. Data either consists of kinesthetic information in the form of n-dimensional vectors or of tactile information in the form of matrices. The data format, which can be a combination of both, is specified at compile time for each application flow. All data is sent on a fixed time base between endpoints, e.g., every 1 ms. Listing 10.1 example shows such a definition in OVGU-HCF. The `ctrlData` structure defines the message format from master to follower, while the `sensData` defines message format for the according feedback. The example listing shows vectors of floating-point values of a given size that are sent in both directions. The content can be freely specified from regular C data types. Thus, it can also consist of video and audio data, for example. The C structs are packed using a marshaling library to be used as payload of either UDP or CoAP [128] messages. The marshaling library

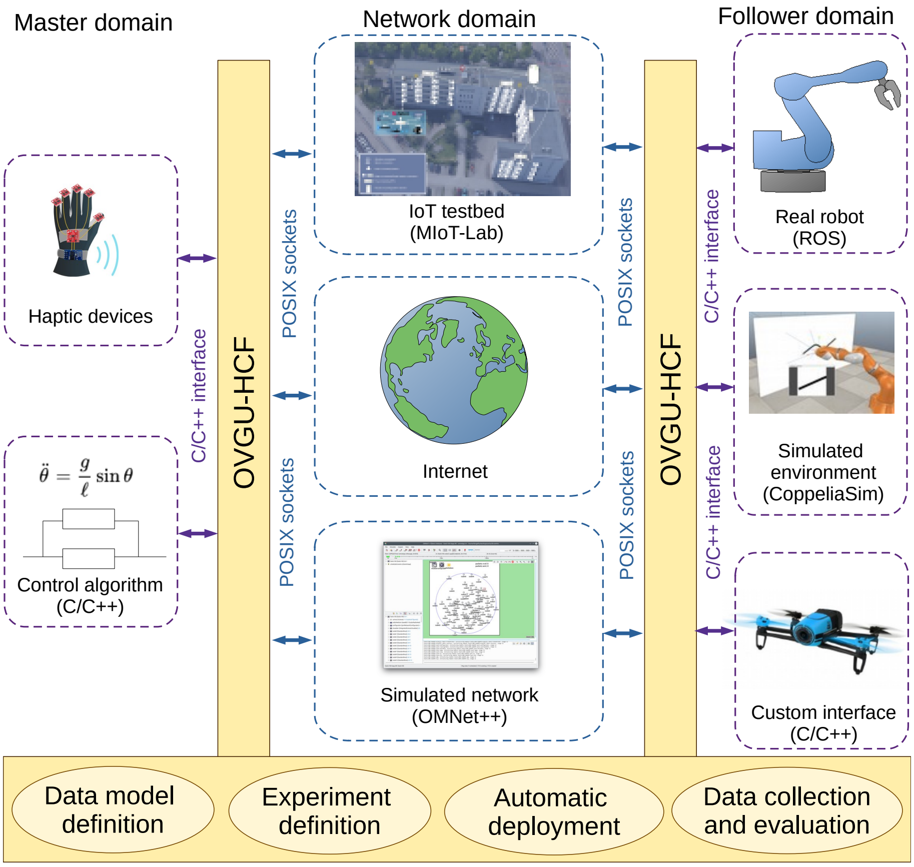


Figure 10.4: Testbed use cases. The experiments can be composed from the shown components from the three domains.

is user-specified. The only prerequisite is that it must be able to convert the message into a C-style byte buffer with starting address and a known (maximum) length.

10.4.5 Networking

The network domain is managed by the MIoT-Lab infrastructure, which provides automatic experiment execution and control. Figure 10.5 shows the distribution of the current testbed nodes within the faculty of computer science. The TBMS controls the experiments and allows them to be represented in DES-Cript [127] format, which potentially allows for recreation on other testbeds that support this format. The DES-Cript file stores every meta-information necessary for experiment execution, which includes, among other things, commands executed on the individual nodes as well as general configurations such as execution times, metadata,

Listing 10.1: Data definition.

```
/* Controller output message */
struct ctrlData {
        /* User specified payload */
        float data[CTRL_DOF];
};

/* Feedback message */
struct sensData {
        /* User specified payload */
        float data[FEEDBACK_DOF];
};
```

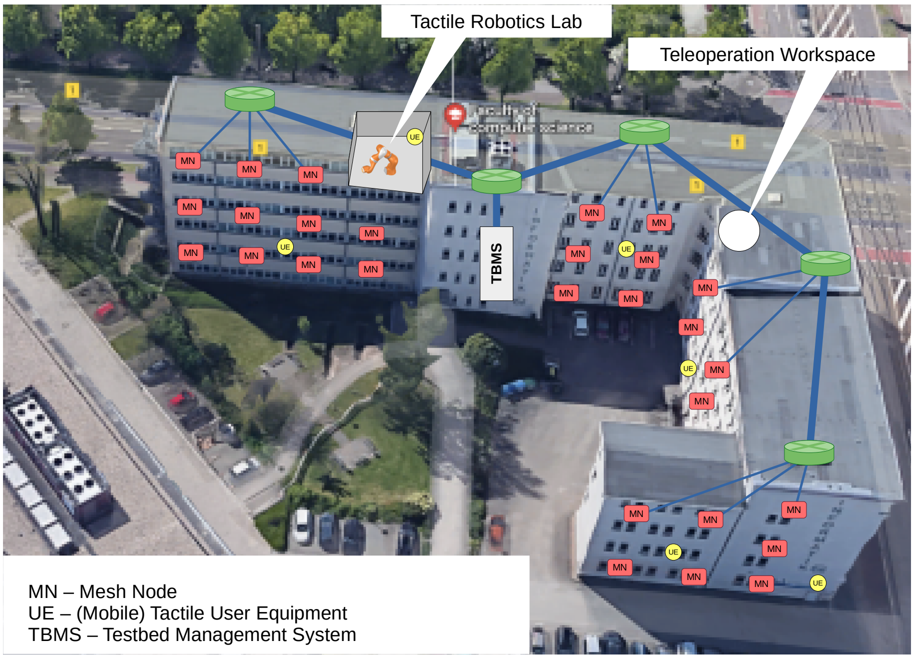

Tactile Robotics Lab
Teleoperation Workspace
TBMS
MN
UE
MN – Mesh Node
UE – (Mobile) Tactile User Equipment
TBMS – Testbed Management System

Figure 10.5: Layout of the OVGU-HC as part of the MIoT-Lab [66, 67].

and initialization scrips for the nodes. The DES-SERT framework [129], in addition, enables the configuration of Layer-3 aspects, especially topology and routing.

10.4.6 Tactile Robotics Lab

The OVGU-HCF can interact with various robotic frameworks, e.g., the Robot Operating System [130], or with custom drivers. We also provide a plugin for the widely used robotic simulation tool CoppeliaSim [124].

A dedicated tactile robotics lab is also part of the OVGU-HC, where workplaces for robotic experimentation exist. A lightweight robot of the type *KuKa iiwa* is ready to be deployed, as well as Quadrotor drones of the type *DJI Mavic* and *ArDrone Bebop*. There is also a teleoperation workspace that consists of a desktop PC equipped with two haptic input devices *3D Systems Touch*.

Instead of real hardware, also simulations via CoppeliaSim can be used as endpoints for experiments. The framework offers of a plugin for CoppeliaSim and a stand-alone application that runs on a remote computer and controls the simulation. The plugin is a shared library and is loaded at the start of CoppeliaSim, allowing access to the over 500 functions of the regular API.

10.4.7 Master Domain Entities

Haptic experiments can be executed either in a *teleoperation mode* using haptic input devices from a teleoperation workspace, or in *control mode* where a control algorithm takes action as a master. Control algorithms are can be implemented freely in C or C++. The framework calls the `step()` function shown in Listing 10.2 at a regular, pre-defined update rate. Controllers like PID, fuzzy, or general AI can be implemented. It is also possible to define interfaces to haptic input devices for experimentation on teleoperation scenarios. In this case, the specified `ctrlData` and `sensData` can be handed over to a driver for the used device.

10.4.8 Haptic Coding

We added support specifically for the task of haptic coding for data compression by manipulating the outgoing (encode) or incoming (decode) datagrams. The coding step is included

Listing 10.2: Controller definition.

```
void step(struct ctrlData &robCtrl,
               struct sensData &robSens)
{
        /* simple P controller */
        robCtrl.data[0] = K_P*robSens.data[0];
}
```

Listing 10.3: API for haptic coding.

```
void decode(char* buffer, char* bufferDecoded,
            size_t size, size_t &sizeDecoded)
{
        /* User specified decoding */
        memcpy(bufferDecoded, buffer, size);
        sizeDecoded = size;
}

void encode(char* buffer, char* bufferEncoded,
            size_t size, size_t &sizeEncoded)
{
        /* User specified encoding,
           setting sizeEncoded to zero
           prevents a datagram from being sent */
        memcpy(bufferEncoded, buffer, size);
        sizeEncoded = size;
}
```

in the framework as a module right before a message is handed to the network controller of a particular endpoint. On the respective peer endpoint, a corresponding decoding step is prepared. Code stubs for coding and decoding can be user-specified. For example, a perceptual deadband filter [6] can be implemented by comparing the current message in the flow to the last one that was sent and applying custom thresholds for sending. The example of Listing 10.3 shows the coding and decoding stubs with a simple pass-through filter that does not perform any data manipulation. The resulting message size is returned by the encoding function. Setting the `sizeEncoded` value to zero results in a message that is suppressed from being sent. It is intended for schemes where message omissions are part of the compression strategy. The message body can be filtered as well, for example for payload compression, stream coding, or even for network coding.

10.5 Testbed Evaluation

We conducted a simulated case study of a 6-DoF robot arm following a line to evaluate OVGU-HCF. We are interested in the maximum transmission frequency that we can achieve on our experimentation computer to show that OVGU-HCF is capable of handling high frequencies of 1 kHz on its own. In a second experiment, we measure the overhead of OVGU-HCF in terms of precision and accuracy.

We investigate a line-following robot as a case study. The experiment setup is as follows. A horizontal line is plotted on a wall as shown in Figure 10.6. A 6-DoF robot in front of the wall starts with its tool center point at one end of the line. The robot movies with a constant horizontal speed of 0.5 m/s towards the other end, following the shape of the line. A camera mounted on the tool center point keeps track of the deviation from it. The line following is achieved with a simple proportional controller, which adjusts the vertical position of the robot. To have an endless scenario, the robot always moves back and forth in the horizontal direction when reaching an end of the line. This horizontal movement is performed locally on

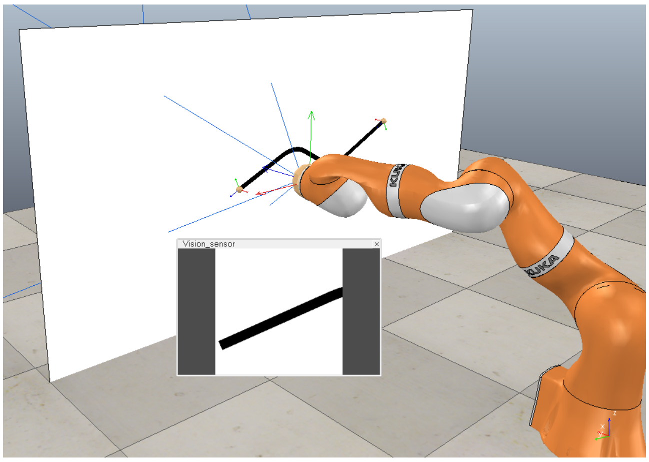


Figure 10.6: Experiment setup in CoppeliaSim showing line following robot and the image obtained from its vision sensor.

the robot. The controller, which is controlling the robot over a network interface, only makes its corrections in the vertical direction. The data flow in this experiment is constituted as follows. From the robot, a vertical strip from the vision sensor's pixels is sent to the controller, which is an array of 256 `uint8_t` values. The controller extracts the positioning error in pixels and computes a scalar horizontal correction value that is sent back through the OVGU-HCF.

We ran the simulation on an Intel Xeon(R) CPU E3-1230 v3 with 8 GB RAM and Ubuntu Linux 18.04.2 LTS (Kernel 4.15.0-58-generic, 64-bit) operating system. For comparability reasons, we do not involve any actual network communication. Instead, we run the experiments locally on the machine with sockets used only for inter-process communication. This way, we see the overhead created by OVGU-HCF, which we aim to keep as small as possible.

10.5.1 Maximum Transmission Frequency

For the experiment, we screen different communication frequencies between the endpoints and compare the achieved with the desired ones. With this baseline data set without network communication involved, the maximum transmission frequency is only limited by the CPU and RAM performance. In addition, the simulation itself runs on the same machine and competes with the framework tasks. Our expectation is that this baseline setup will support frequencies far beyond 1 kHz to show that the introduced overhead on CPU and RAM is minimal.

Figure 10.7 shows the results at different frequencies up to 600 kHz. The results differ on both the sending and the receiving ends. The packet reception on the receiving end goes into saturation much earlier, showing that packets have to be dropped by the kernel due to overrunning buffers. The overall performance is therefore indicated by the receiving curve. Here the system goes into saturation at a frequency of about 350 kHz. As this is far more than the required 1 kHz, the computational overhead of the framework is small enough to be no limiting factor compared to the networking overhead in a real experiment.

10.5.2 Impact on Precision and Accuracy

We measure the accuracy of the robot by the cumulated pixel error over the experiment duration (2.5 s for a traversal from one end of the line to the other), and precision by the standard deviation of the same value over several experiment runs. One pixel of error is equivalent to about 1.5 mm of error between Transmission Control Protocol (TCP) and the line. We also include a "local control" solution that is purely implemented in CoppeliaSim without using OVGU-HCF at all. This local control solution does therefore not include overhead of inter-process communication through sockets, and we can compare the impact of the framework on the precision and accuracy. Measuring the achieved precision and accuracy

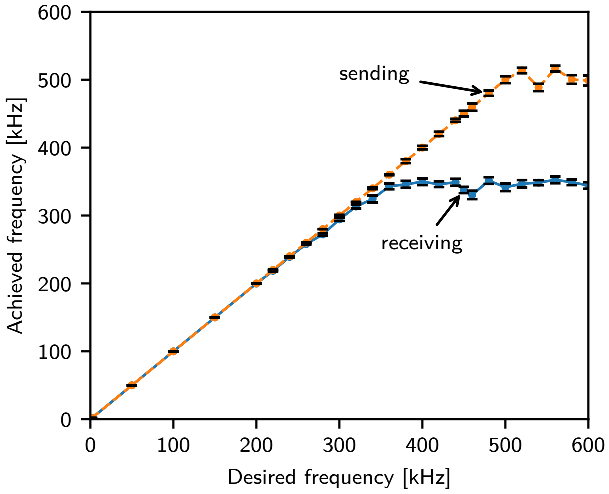


Figure 10.7: Desired vs. achieved update frequency on a local machine ($n = 250, \alpha = 0.01$).

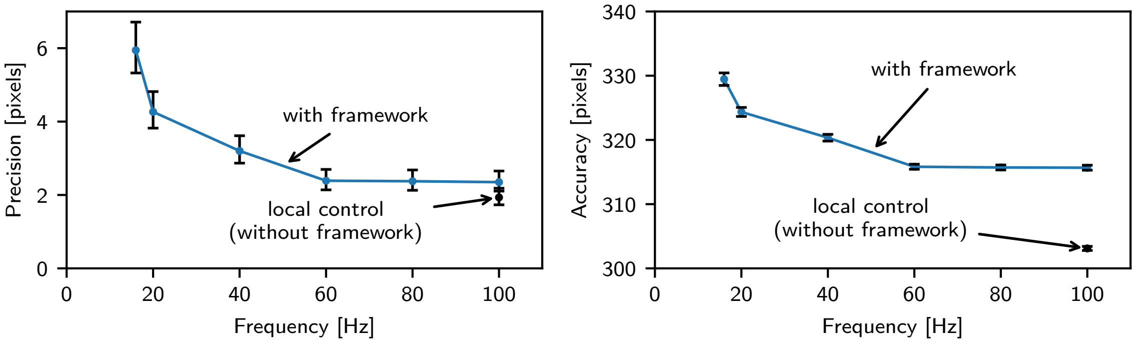


Figure 10.8: Influence of update frequency on precision and accuracy ($n = 250, \alpha = 0.01$).

of the local control is expected to show no different behavior than that of the implementation within OVGU-HCF. The simulator supports a minimum time resolution of 10 ms, so we settle the update frequency of OVGU-HCF to 100 Hz in this case and compare results.

The results for precision are shown in Figure 10.8. Its precision of the local control lies within a confidence interval of [1.73, 2.18] pixels ($n = 250$, $\alpha = 0.01$), with estimator (sample standard error) 1.93 pixels. OVGU-HCF achieved a precision within a confidence interval [2.10, 2.65] pixels, with estimator 2.35 pixels.

For local control, the cumulated error of the robot is 303.10 ± 0.32 pixels ($n = 250$, $\alpha = 0.01$), whereas with OVGU-HCF at 100 Hz update rate the error was 315.67 ± 0.39 pixels. The resulting discrepancy of 4.1 % needs to be addressed by application developers.

At 100 Hz, the precision of the local control and the OVGU-HCF implementation overlap in their confidence intervals, so we conclude that the framework has no significant impact on the precision. For the accuracy, the difference is significant as the confidence intervals do not overlap. However, the absolute difference with 20 pixels cumulated over 2.5 s time is very low compared tho the camera resolution of 800×600 pixels. The framework, therefore, has introduced an overhead resulting in reduced accuracy, but the magnitude is relatively low.

10.5.3 Example Use Case: Choosing an Update Frequency

We also briefly demonstrate the use-case of assessing the impact of communication effects on the given application with OVGU-HCF. This is a use case that can not be investigated with the local control of CoppeliaSim, thus it shows the benefit that can be drawn from using our framework. For this small case study, we vary the update frequency (i.e., the packet sending rate) and measure the effects on accuracy and precision. Both precision and accuracy decrease with decreasing update frequency, as is expected. Due to the reduced message flow, information is lost for the control algorithm, such that it can no longer maintain the previously achieved control quality. The lowest update frequency where we observed the robot being able to complete its task was 16 Hz. At that frequency, the accuracy went down as low as 329.46 ± 0.98 pixels (cumulated error). The precision, however, shows an even higher sensitivity to different update frequencies. The results are shown in Figure 10.8.

As a result of such an experiment, an application engineer can now assess what frequency to choose for the given system. The results show that a higher frequency yields better quality, but also the network load increases so that it might be worthwhile to reduce the frequency as much as possible. Typically, an engineer has a requirements catalog for the application, specifying a required minimum precision and accuracy. The engineer would now choose the lowest possible update frequency that still fulfills the requirements. The framework now serves as a validation and evaluation tool, and the experiment results can be stored for documentation of the design choice.

CHAPTER 11

Evaluation

In this chapter, we evaluate our contribution in two parts. First, we evaluate the Tactile Coordination Function in the MIoT-Lab, showing both that its implementation on real hardware is possible, and it contributes to Haptic Communication in WiFi multi-hop networks. Second, we evaluate our probabilistic latency model in a simulated case study of a large-scale WiFi network. From both evaluations, we can conclude the applicability of our contributions.

11.1 Evaluation of the TCF in the MIoT-Lab

In this case study we use the MIoT-Lab to evaluate the Tactile Coordination Function developed in Chapter 9. As in that chapter we have only relied on a simulation so far, we need to demonstrate that the proposed function can actually be implemented in real WiFi networks. In addition, the real-world data can further prove its effectiveness. The additional evidence brought by this testbed run completes the proof of concept for our proposed amendment.

11.1.1 Hypothesis

The hypothesis of the experiment is similar to the one of the simulation experiment in Chapter 9. A haptic flow transmitted with the TC Access Category should be unaffected by a present background transmission on the same channel. Therefore, we install a multi-hop haptic flow and measure the mean packet latency and the jitter before and after the onset of a background transmission.

(H1) The onset of the background transmission has a significantly smaller effect on the latency with TCF than it has with unmodified IEEE 802.11.

(H1') Regardless if TCF or unmodified IEEE 802.11 is used, the onset of the background traffic has a significant effect on the latency (alternative hypothesis).

11.1.2 Experiment Setup

We utilize 8 WiFi-capable nodes from the testbed which are in mutual communication range, numbered from 1 to 8. The conflict graph of this network is fully connected, meaning that all nodes are in interference range of each other. They are also in communication range between each other, as our measurements suggest, however, we enforce a sparsely connected network topology that is shown in Figure 11.1. Nodes 5, 7, 6, 1, 2 form a linear topology in this sequence where we transmit haptic data on a single haptic flow. We experiment with varying hop counts to observe the behavior within different states of channel utilization. In each case, node 5 is the source of the haptic flow, while either node 7, 6, 1, 2 act as destination node. Thus, the haptic data is transmitted over either 1, 2, 3, or 4-hop paths. At the same time, an additional background flow can be activated between nodes 8 and 3, as shown in Figure 11.1. The background flow is within the interference range of the haptic flow.

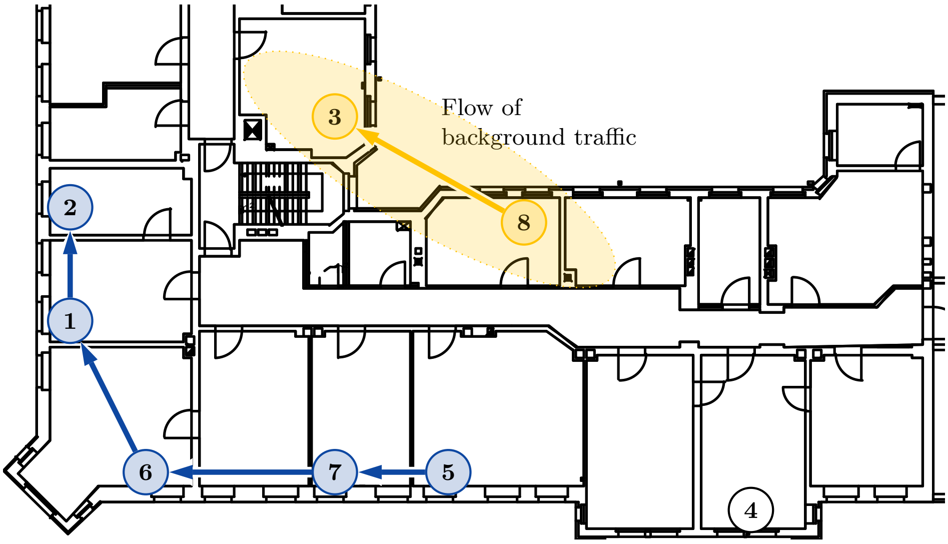


Figure 11.1: Topology of the testbed experiment. The testbed consists of 8 nodes. Nodes 5, 7, 6, 1, 2 form a linear topology and transmit a haptic flow from the source node 5 over a varying number of hops. At the same time, a background flow is active between nodes 3 and 8 on the same channel. The node IDs come from the testbed itself. Node 4 is not participating in this experiment.

Table 11.1: Overview of the experiment parameters. Sets of values indicate variations between different runs, resulting in 16 runs total.

Parameter	Value
Experiment time	300 s
Hop distance	2–5 m
Max. bitrate	600 Mbit/s
Operating System	Linux 4.19.194-3
Driver	`ath_10k/mac80211`
Channel	Ch. 44, 20 MHz

Parameter	Haptic data	Background traffic
Hop count l	{ 1, 2, 3, 4 }	{ 0 (not present), 1 }
Access Category	{ TC, BK }	BK
Payload size s	100 Byte	16 kByte
Sending interval	1 ms	1/60 s
Packets total	300 000	18 000

We conduct the experiments with two prioritization modes. First, we use classic WiFi, with no prioritization used at all. Here, the haptic data and the background data use the same prioritization, which renders their scheduling on the channel a purely random process. In each contention phase, either the haptic data can win the competition for the first transmission, or the background data. In a second set, we use the WiFi TCF amendment as described in Chapter 9 and prioritize the haptic data by labeling it with the proper Access category. In this set, the haptic flow should always access the channel first in all contention phases.

A total of 4 × 2 × 2 experiment runs are conducted, depending on the variations in hop count (1 to 4), the activation of the background traffic (*background* or *no background*), and the used prioritization (*classic* or *TCF*).

The NIC parameters and the traffic configuration of the experiment is summarized in Table 11.1.

11.1.3 Implementation Details

The implementation of TCF on a real system is not trivial as most drivers only support a fixed number of 4 EDCA queues. The hard limit comes from the original WiFi standard that defines exactly 4 queues. The queues are implemented either in hardware or in the proprietary firmware of Network Interface Cards (NICs), leaving no room for extension. However, some NICs allow free configuration of the transmission parameters, which allows for repurposing the queues for different Access Categories. We configure them to reflect the four Access

Categories TC, VO, VI, BE, in that order. The parameters are given in Table 9.1. The BK Access Category has been dropped. As we only plan to use two different Access categories in our experiment, this is a feasible implementation. The queues do not have any particular semantics beyond the attribution of the transmission parameters.

The Testbed nodes run a Linux operating system with a SoftMAC driver architecture. SoftMAC drivers rely on the Linux implementation of the MAC Layer, which is contained in a module called `mac80211`. It allows for free configuration of the EDCA queues in conjunction with certain drivers, such as the `ath_10k` used for our NICs. In the Kernel, however, we had to make a minor code change to allow the EDCA parameters to be changed in the IEEE 802.11 mesh mode. The code change is depicted in Listing 11.1.

Listing 11.1: Changes made to the function `nl80211_set_wiphy()` in `net/wireless/nl80211.c` of Kernel version 4.19.194-3.

```
static int nl80211_set_wiphy(struct sk_buff *skb, struct genl_info *info)
{
        ...

        if (info->attrs[NL80211_ATTR_WIPHY_TXQ_PARAMS]) {
                struct ieee80211_txq_params txq_params;
                struct nlattr *tb[NL80211_TXQ_ATTR_MAX + 1];
                if (!rdev->ops->set_txq_params)
                        return -EOPNOTSUPP;
                if (!netdev)
                        return -EINVAL;
                /* the following three lines had to be commented out */
                /*
                if (netdev->ieee80211_ptr->iftype != NL80211_IFTYPE_AP &&
                    netdev->ieee80211_ptr->iftype != NL80211_IFTYPE_P2P_GO)
                        return -EINVAL;
                */
        }

        ...
}
```

The traffic in the experiments was generated with the tool `iperf`. It is a client-server application that offers the generation of isochronous traffic, which is marked automatically to measure various metrics, such as throughput, latency, jitter, and packet loss. The clocks of all nodes are synchronized with the Precision Time Protocol (PTP). PTP was running as a daemon during the whole set of experiments and synchronized the hardware times via the wired network infrastructure dedicated for experiment control communication. The clock of node 1 was used as a master clock. The offset of the nodes was below 100 µs.

11.1.4 Experiment Results

Figure 11.2 shows the latency measurements from the experiment. The mean and standard deviations from all 16 runs are also shown in Table 11.2.

Without background transmission, both the classic channel access and TCF behave basically identical. As no other traffic besides the one haptic flow exists in the network, this is expected, because contention on the medium only takes place between the individual single-hop transmissions belonging to the haptic flow. With the onset of the background traffic, the latency distributions for the haptic data increase both for the classic scheme and the TCF scheme. However, the prioritization with TCF shows a much more graceful increase in the latency distribution compared to the classic scheme. It shows that for the 3-hop case, the haptic flow latency could meet the 1 ms mark with TCF, while it clearly misses this mark with the classic scheme. In the 4-hop case, the mean latency exceeds even the one second mark with the classic scheme, which can be considered as a collapse of the network service quality. With TCF, however, the 4-hop case could provide a mean latency of at least under two milliseconds, which would still be a valid service quality for many Haptic Applications.

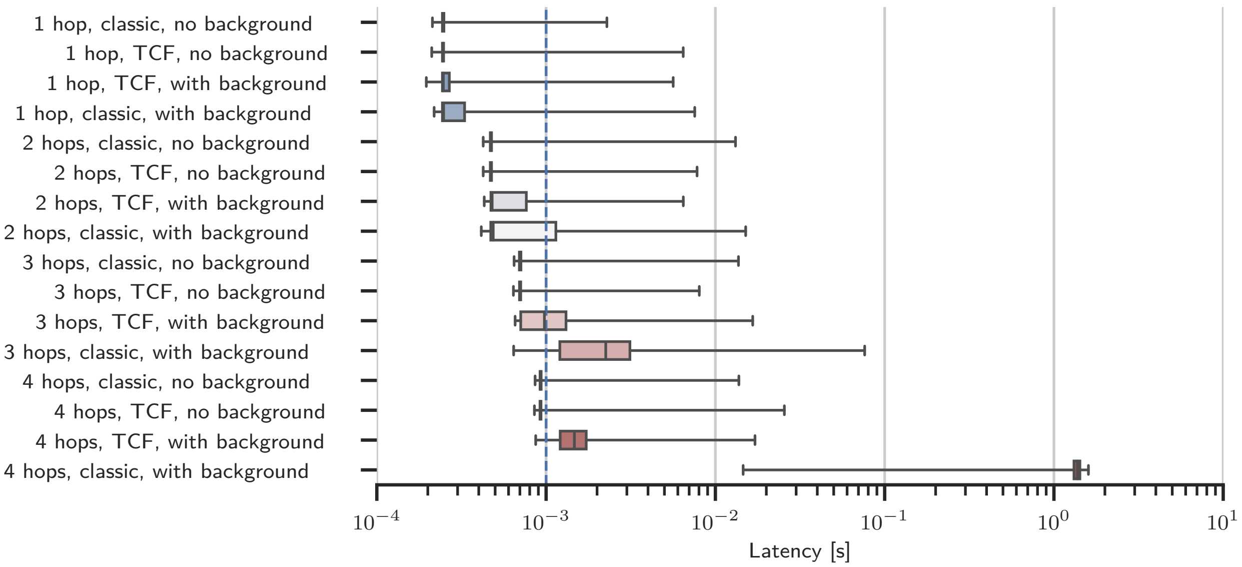


Figure 11.2: Latency measured for the haptic data from source to sink during the experiment runs (n=300 000).

Table 11.2: Measured mean latency and standard deviation from the testbed experiment ($n = 300\,000, \alpha = 0.001$). All values given in milliseconds.

Dataset	Mean	Std-dev
1 hop, classic, no background	0.248 00 ± 0.000 12	0.020 30 ± 0.000 09
1 hop, TCF, no background	0.247 00 ± 0.000 16	0.026 65 ± 0.000 11
1 hop, TCF, with background	0.293 00 ± 0.000 61	0.101 99 ± 0.000 43
1 hop, classic, with background	0.345 00 ± 0.001 26	0.210 47 ± 0.000 89
2 hops, classic, no background	0.477 00 ± 0.000 31	0.051 96 ± 0.000 22
2 hops, TCF, no background	0.475 00 ± 0.000 24	0.040 38 ± 0.000 17
2 hops, TCF, with background	0.621 00 ± 0.001 30	0.216 58 ± 0.000 92
2 hops, classic, with background	0.861 00 ± 0.003 31	0.550 67 ± 0.002 34
3 hops, classic, no background	0.712 00 ± 0.000 63	0.105 36 ± 0.000 45
3 hops, TCF, no background	0.711 00 ± 0.000 60	0.099 19 ± 0.000 42
3 hops, TCF, with background	1.049 00 ± 0.002 44	0.406 11 ± 0.001 73
3 hops, classic, with background	2.542 00 ± 0.020 72	3.448 81 ± 0.014 65
4 hops, classic, no background	0.948 00 ± 0.000 93	0.154 56 ± 0.000 66
4 hops, TCF, no background	0.943 00 ± 0.001 74	0.289 05 ± 0.001 23
4 hops, TCF, with background	1.511 00 ± 0.002 84	0.473 15 ± 0.002 01
4 hops, classic, with background	1291.160 00 ± 1.619 00	269.491 92 ± 1.144 81

11.1.5 Discussion

The experiment shows the necessity of a new Access Category dedicated to Haptic Communication to be established in IEEE 802.11. The prioritization scheme provided with TCF shows feasible results, especially in the 4-hop test case, where a reasonable network service could only be provided with TCF. Unfortunately, however, TCF is not fully capable to prioritize haptic traffic over other Access Categories in any case, which results in a small latency increase of the prioritized data as soon as background traffic sets in. With an ideal prioritization, the onset of background traffic would not affect haptic flow performance at all. It is unlikely to achieve such behavior with just the adaption of EDCA parameters, as we chose to with TCF. Such behavior would require stricter means of priority enforcement, such as a polling mechanism, or even dedicated slots for haptic transmissions, however, such means have not proven to be effective in WiFi networks in the past.

The benefit of TCF is that it is compatible with each of the established channel access mechanisms used in WiFi. More specifically, TCF is backward-compatible to the original IEEE 802.11 standard. TCF-capable nodes can coexist with non-TCF nodes on the same channel, and the function could function as usual, while the prioritization of the TCF-enables haptic flows would still be able to be prioritized in the network. This is possible as the experiment shows that the onset of the background traffic, which does not use any TCF extension, does not deny the haptic flows at all, but only increases the delay due to more contention.

11.2 Simulation Results for the ME Queueing Model

In this section, we evaluate the ME queueing model for latency prediction in a more realistic scenario than with those used in Chapters 7 and 8. Additionally, we demonstate the combination of both models to predict both the single-hop delays and the queueing delays. We use a simulation, as it can be easily replicated and modified. The case study includes several concurrent multi-hop flows that share medium time according to the rules of plain and unmodified IEEE 802.11. Our prediction model shows accurate in this scenario, as the results indicate.

11.2.1 Experiment Setup

We chose a confined, two-dimensional area of 500×500 m and placed a number of $n_{\mathcal{V}} = 100$ nodes randomly in that area. From these nodes, only some actively transmit data as part of a running application while the majority is just passive, as is a typical situation in productive WiFi networks. In addition, we vary the number of active flows to investigate the accuracy of our model in various utilization scenarios. The network is depicted in Figure 11.3. Four flows, labeled A, B, C, and D, are defined, with their flow paths highlighted in Figure 11.3. We conduct four experiment runs in total, with each run activating one additional flow in order to increase the network load step by step. Thus, flow A is active in all four experiment runs, B in runs two to four, C in runs three and four, and, finally, flow D is only active in the fourth run. So with each run, the load gradually increases, with the medium utilization in the final run reaching approximately 20 %. The flow parameters are given in Table 11.3. All flows are unidirectional. For example, A1 transmits to A2, B1 to B2, and so on. The passive nodes are simulated in OMNet++, and they contribute only an insignificant amount to the overall network utilization. For example, they send regular WiFi beacons, but they do not actively send any data on the Application Layer.

Figure 11.3 also shown the conflict graph of the network. It is not fully connected, and thus some of the nodes are able to transmit simultaneously without affecting each other. This constellation is a strength of CSMA-based networks which can leverage the effect of spatial reuse. This effect can be reflected by our model.

In this somewhat larger simulation scenario, the influence of the queueing delay is higher compared to the small-scale Butterfly and FourCross networks that we used before. It can not be easily modeled by a binomial distribution as in Chapter 7, and its overall influence on the latency is significant, as the expected queue length can reach up to two packets. Due to the conflict graph not being fully connected, a model based on a Gamma distribution is not sufficient, as the model parameters must be estimated for each node individually. The ME queueing model resolves the necessity for parameter estimations, and the number of model parameters is low.

11.2.2 Hypothesis

The prediction accuracy can be determined by the amount of overlap between the realized latency histograms and the predicted PDF. With increasing load, the realized latency must increase, too, and this circumstance should be reflected by our prediction. Therefore, we have the following hypothesis to test:

(H2) As the number of flows increases in each experiment run, the prediction from the ME model should reflect the realized latency distribution of all active flows. This should show in overlapping latency histograms and predicted PDF of the path delay.

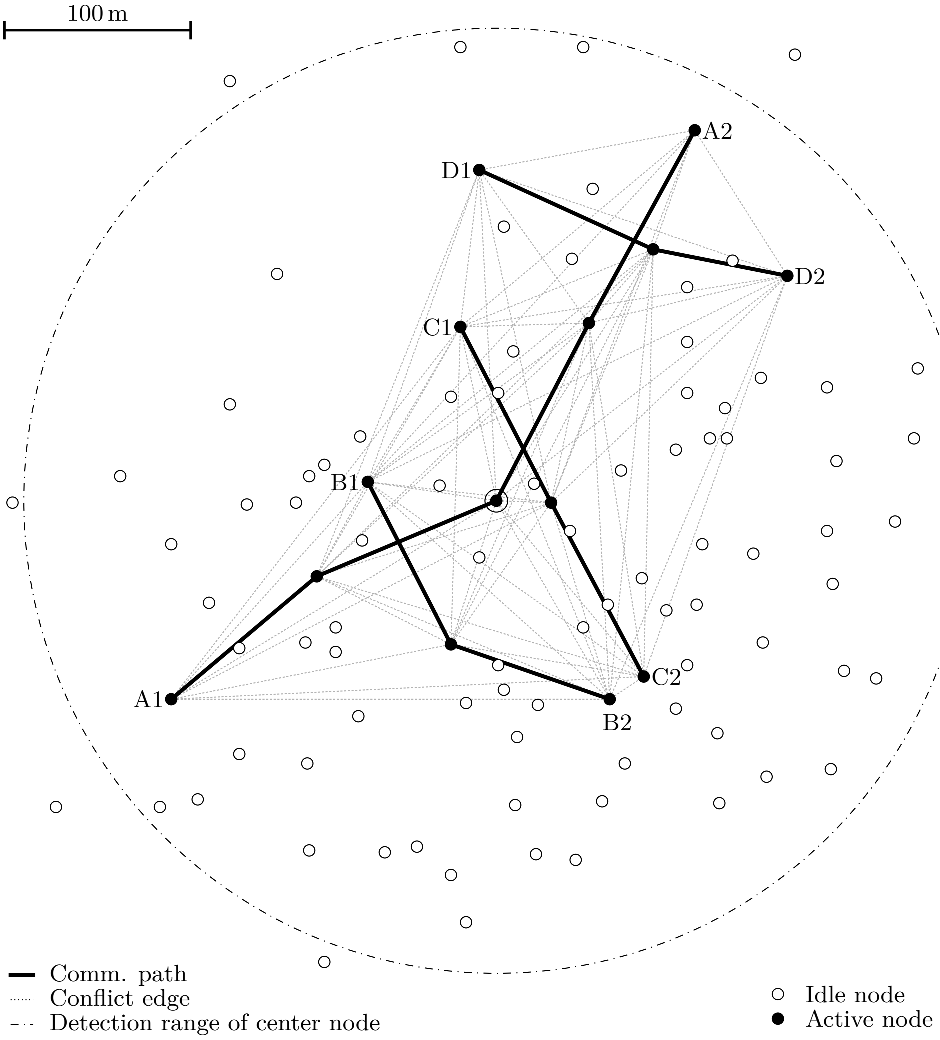


Figure 11.3: Topology of the simulated WiFi network of which we want to model the latency behavior with our ME queueing model. Four flows, A1→A2, B1→B2, C1→C2, and D1→D2, are started one after another, beginning with flow A. With all four flows active, the network utilization at the highlighted center node reaches 20.1 %.

(H2') The prediction underestimates the latency distribution significantly with increasing network load.

It should be noted that the prediction cannot significantly overestimate the real latency. The main factor that we consider in this experiment is the influence of the medium utilization and the resulting queueing delay, which can only increase the realized latency, but not reduce it. We already have shown the basic accuracy of our model in Chapter 7 where we predicted the latency distribution for smaller-scale networks.

11.2.3 Prediction Model

For latency prediction, we combine the two prediction models from Chapters 7 and 8. The queueing delay is modeled as a random variable Q_v for each node v as elaborated in Chapter 8, using Algorithm 3. The queueing model takes the network topology and the conflict graph, which we are able to determine for the given scenario. The relative load load(v), which is another required input for the ME model, is measured in the OMNet++ simulation. The required calculations are conducted according to Algorithm 3 on a PC equipped with a 8-Core CPU and 16 GB RAM. The PDFs of the queuing delay are then given as discrete ordered lists, assigning a probability to each transmission slot time.

Table 11.3: Simulation parameters.

Parameter	Value
Number of nodes $n_{\mathcal{V}}$	100
Coverage area	$500 \times 500\,\mathrm{m}$
Radio Model	INET `Ieee80211ScalarRadioMedium`
Node model	INET `StandardHost`
NIC Model	INET `Ieee80211Interface`
Routing	Static, with pre-initialized ARP caches
NIC properties	IEEE 802.11ac, 4×4 MIMO
TX power	2 mW
Channel	20 MHz bandwidth @2.4 GHz, free space path loss
Flow message length	100 B
Flow send interval	5 ms
Number of flows n_{f}	1 to 4 (A1→A2 to D1→D2)
Number n of packets sent	120000
Resulting medium utilization	8.3 % with 1 active flow
	12.5 % with 2 active flows
	16.4 % with 3 active flows
	20.1 % with 4 active flows

Table 11.4: ME model parameters.

Parameter	Symbol	Value
Data rate	r_v	300 Mbit/s
Packet error rate	p_{per}	0.001
Subnetwork affinity	τ	0.09

Table 11.5: Related constants from IEEE 802.11.

Constant	Symbol	Value
Contention window	$CW_{\min}, CW_{\max}$	$15, 1023$
Slot/IFS times	t_{slot} / t_{ack} / t_{AIFS}	9/18/36 μs
Header data rate	r_{base}	6 Mbit/s
Header size	h	320 bit
Max. retransmission count	$\hat{m}$	7

To retrieve the single-hop delay, the queueing delay Q_v, measured in time slots, must be multiplied with the duration of transmission from node v to u, $D_{v;u}$. The path delay $T_{\mathcal{P}}$ for a flow path $\mathcal{P}$ is then determined by adding up all single-hop delays:

$$T_{\mathcal{P}} = \sum_{(u,v)\in\mathcal{P}} D_{u;v} Q_u.$$

The busy time $D_{u;v}$ can be determined by the method in Chapter 7. As all values are random variables, the operations presented in Section 6.4.2 for summation and multiplication must be used to combine the individual PDFs to retrieve the final PDF of $T_{\mathcal{P}}$.

The described process requires a set of parameters, which mostly are hardware parameters that can be taken from the used IEEE 802.11 protocol specification. these are shown in Table 11.5. Additionally, a smaller set of parameters can not be settled simply and can be considered as true model parameters. They need to be fitted carefully to the scenario. These are the packet error rate p_{per}, the data rate r_v, and the affinity τ for a node to change its regime from On to Off, or vice versa. These must be fitted carefully to the application. The fitting set for our scenario is given in Table 11.4.

11.2.4 Experiment Results

The results of the experiment is shown in Figure 11.4. The plotted histogram data is the observed distribution of the latency through the experiment runs and for the individual flows. The corresponding expectations are shown as blue curves, which represent the respective PDF of the path delay for the particular flow (A, B, C, or D). An ideal prediction would match the observed distributions. At least for the flows A, B, C, in all the runs, this is the case – granted a small over-estimation, especially with the run with lower utilization. For flow A, however, the prediction is strongly over-estimated by our model. For the same reasons as in the evaluation in Chapter 7, the expectation has a bias. We do not consider over-expectation of the shown amounts as a flaw. Typically, for the application of the ME model as a means of resource assessment, over-estimation is not a concern, as no resources would be assumed which could not be provided by the system.

As the model does not under-estimate the observed latency distribution, we accept the primary hypothesis **H2**.

Table 11.6: Observed median latency and jitter in ms.

	A1→A2		B1→B2		C1→C2		D1→D2	
n_f	lat.	jit.	lat.	jit.	lat.	jit.	lat.	jit.
1	0.730	0.099						
2	1.032	0.226	0.604	0.336				
3	1.213	0.400	0.659	0.511	0.841	0.281		
4	1.385	0.612	0.822	1.179	1.040	0.536	1.029	0.579

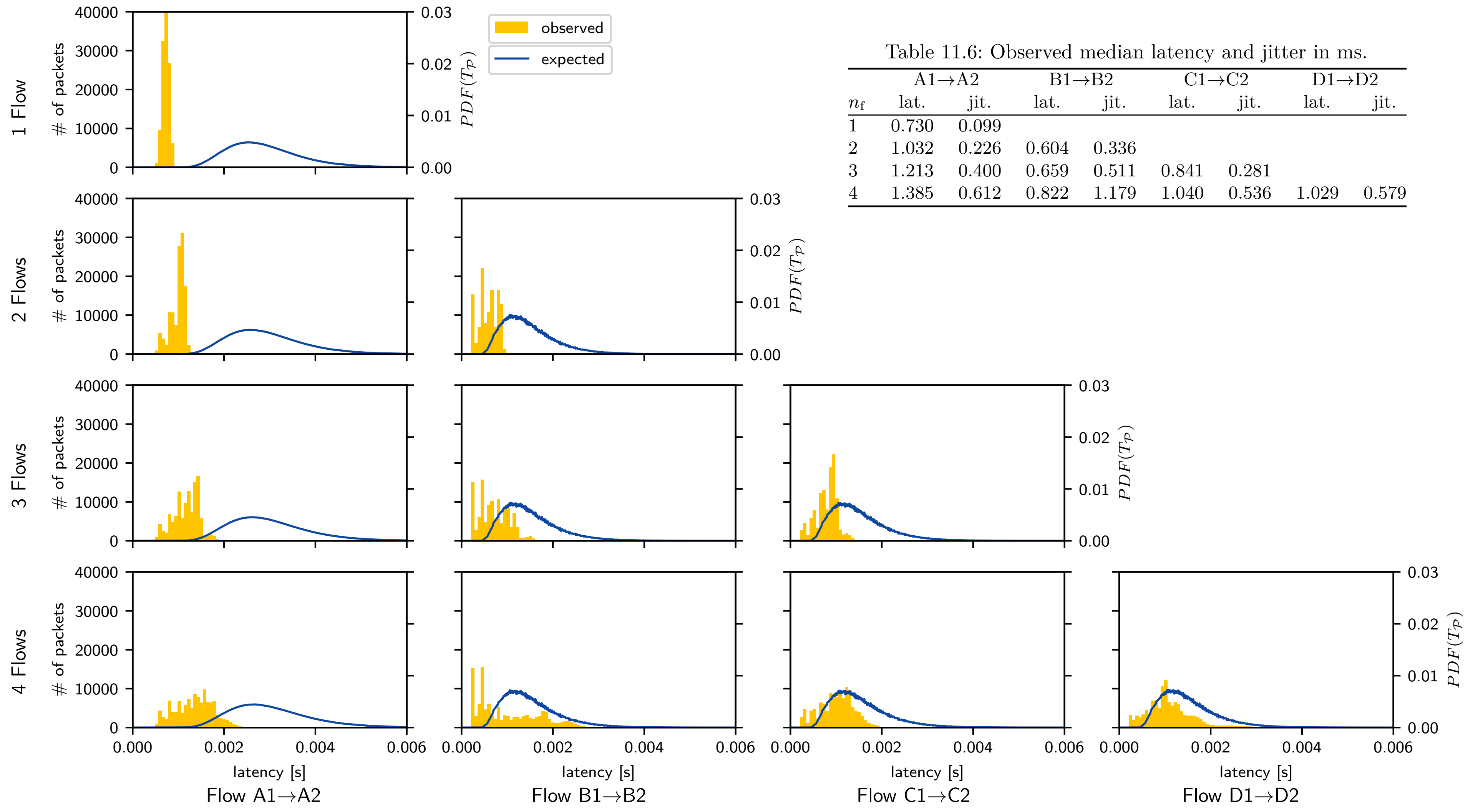


Figure 11.4: Expected vs. observed latency distribution in the examined scenario ($n = 120\,000$). Four experiment runs were conducted in a simulated 100-node WiFi network, with one additional flow activated with each run.

CHAPTER 12

Discussion of Results

With this thesis, we have investigated three major aspects of the upcoming Tactile Internet: The aspect of an application framework, the modeling of latency and reliability, and MAC-Layer protocol design for Haptic Communication. We have provided a step towards the realization of Tactile Internet Applications, with the key design goal to provide worldwide Haptic Communication and an 'Internet of Skills'. All these aspects are based on the assumption of an open Tactile Internet architecture that makes intense use of Edge Cloud technologies and can be assumed to have a two-plane structure. In this architecture, the load of URLLC services is carried by the Access Networks, and as a conclusion drawn from this thesis, the distinction between Access Networks and the Backbone is necessary. Digital Twins, Artificial Intelligence and Edge Cloud technologies act as mediators between these two network roles, and finally allow to overcome the 1 ms-problem that is induced by the laws of physics.

Our three-fold contribution shows how the worldwide Tactile Internet can be realized while still respecting the existing Internet infrastructure and preserving its current mode of operation as a collection of independent Autonomous Systems. This is a key point, as changes and improvements to the Internet are not made by a single, central authority, but by a large community of stakeholders, service providers, and users. A transition to the Tactile Internet must come iteratively, and might require to operate with mostly existing technologies in a first design iteration.

With the application framework HDTF, a software architecture for the Tactile Internet is defined that can solve the problem of worldwide Haptic Communication. Its underlying distinction between the Access Network plane, which is responsible for URLLC services, and the Core Network plane that cares for Digital Twin synchronization is an essential cornerstone. With this distinction, and the resulting division of software into four different components, master endpoint, master Digital Twin, follower Digital Twin and follower endpoint, the 1 ms-problem can be solved efficiently and by proper technical means compatible with the current layout of the Internet. We have shown the applicability of HDTF with the realization of the fundamental design concepts in the OVGU-HC, a testbed for Haptic Communication at the Otto-von-Guericke-University of Magdeburg, as part of the MIoT-Lab. Although further development is necessary to provide concrete software libraries, especially in the field of Digital Twins and proper prediction models, the HDTF can act as a guideline for future software platforms for the Tactile Internet.

The probabilistic latency and reliability model for Access Networks, based on our ME queueing model, is a novel modeling technique for URLLC. Such modeling is necessary for new networks, as latency and reliability become more and more a concern for the future Internet. Proper modeling is a key requirement for the planning and conceptualization of new network infrastructure, and modeling techniques for wireless multi-hop networks providing URLLC services have been in a very early stage previously. With the ME queueing model, we contribute to the understanding of wireless networking and the relation of network utilization and the required performance metrics.

With the Tactile Coordination Function for IEEE 802.11, finally, we contribute methods for integrating Haptic Communication in existing wireless technologies. For WiFi multi-hop networks, we have shown their capacity to provide the required URLLC services. The TCF amendment that we have proposed in this thesis can be implemented in IEEE 802.11 while respecting the major design challenge of backward compatibility with older WiFi standards. This ensures a maximum interoperability between devices, both with legacy and current networks. Moreover, the proposed changes can be implemented in pure software for hardware including IEEE 802.11ac (WiFi 5), which includes the last two WiFi generations as of the printing date of this thesis.

Our approaches into all three directions place a further guidestone on the path to the iterative realization of the Tactile Internet. We believe that all three steps, from low-layer protocol development, to modeling, to a high level software framework must be taken together in order to achieve the goal of worldwide 1 ms-connectivity. Improvements in each of the aspects also influence the other two, leading to a step-by-step improvement, as shown in the following Figure 12.1.

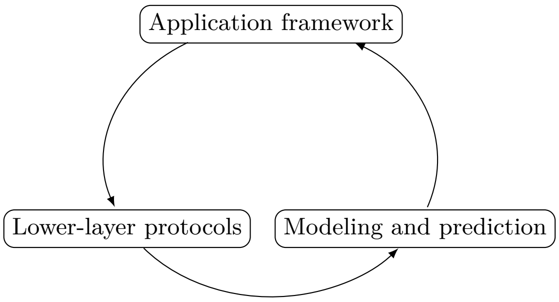


Figure 12.1: Iterative development process for advances on the Tactile Internet.

12.0.1 Future Work

Future research can extend all of our three particular contributions.

The ME queueing model has the evident drawback of its computational complexity. The set of phases grows exponentially, as the complex interactions between each pair of nodes must be taken into consideration for the network utilization that leads to the final queuing model. Although the set of valid phases to consider can be much smaller, each phase has to be checked for validity individually. The construction of the phases can be done offline, assuming an otherwise static network, and then be re-used as long as no new node enters the network. The size of the set, however, is not limited, and is bigger the less dense a network is. Large-scale networks with low overlap in the nodes detection ranges are computationally more intense. Although we have managed to apply the ME queuing model to a fairly large-scale network, the question of simplifications arises to support its applicability in practice. A candidate for a simplification can be, again, the traditional approximation by Gamma distributions. As seen in our experiments, the resulting estimate PDFs could often resemble Gamma distributions. However, the estimation of the two utilization parameters of the Gamma distribution, which must be made for each node, leads to a large parameter set. A solution could be to develop the ME model to yield just the utilization parameters instead of full PDFs.

Another direction for further research should be the appropriation of the HDTF. The lack of best practices and real example applications makes it hard to specify a framework in enough detail to start any implementation at the current level of technology. While some subtasks, for example for Haptic Communication, are clear enough to implement, especially the requirements for the Digital Twins, their level of detail, and the represented aspects remain unclear and need more research and development. Also, the synchronization of Digital Twins will be a research aspect. The question remains if a Digital Twin modeling language can finally cover all use-cases, so that a general approach can be made. The same is true for the storage techniques that can be used. We believe that specialized solutions for a few niche applications will emerge first. Telesurgery applications are promising candidates as well-trained surgeons are among the most valuable of all professionals who rely on physical interaction.

As the Tactile Internet is one application for URLLC, the development will affect future communication networks, such as 6G. A major concern will be the introduction of multi-hop characteristics. The effort is subsumed under the keyword Self-Organizing Network (SOM). Latency and reliability models are a contribution to the development, as cellular networks are relying on proper resource allocation and must be optimized for cost-effectiveness. Although the development SOMs does not seem to be a priority at the moment, as it has been on the agenda of 3GPP for several years, the increasing device density will require effective, self-organizing multi-hop networking.

Finally, the Tactile Internet will require new Application Layer protocols dedicated to kinesthetic or tactile data. We have not covered this topic at all in this thesis, as the nature of data is not too clear yet. However, we have already contributed a Wavelet-based coding scheme [131] for kinesthetic data. The IEEE 1918.1 Work Group has addressed the issue with a preliminary Application Protocol, but we believe that community-driven protocols will quickly emerge as an alternative. Such protocols could be based on similar IETF-standardized alternatives, such as RTP or the Quic Protocol. It is likely that UDP-based protocols will be used for the Haptic Communication, as TCP has too much overhead, and the reliability mechanisms will not be useful on top of URLLC network infrastructure. We will continue our own research with the adaptation of existing open protocols from IETF for Haptic Communication, We believe that protocol development will be the next major step towards a better understanding of the future software architecture of the Tactile Internet.

Bibliography

[1] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo. A survey of non-orthogonal multiple access for 5g. *IEEE Communications Surveys Tutorials*, 20(3):2294–2323, 2018.

[2] D. Gesbert, M. Shafi, D. shan Shiu, P. Smith, and A. Naguib. From theory to practice: an overview of MIMO space-time coded wireless systems. *IEEE Journal on Selected Areas in Communications*, 21(3):281–302, apr 2003.

[3] F. H. Fitzek, S.-C. Li, S. Speidel, and T. Strufe. Chapter 1 - tactile internet with human-in-the-loop: New frontiers of transdisciplinary research. In F. H. Fitzek, S.-C. Li, S. Speidel, T. Strufe, M. Simsek, and M. Reisslein, editors, *Tactile Internet*, pages 1–19. Academic Press, 2021.

[4] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis. 5G-Enabled Tactile Internet. *IEEE Journal on Selected Areas in Communications*, 34(3):460–473, March 2016.

[5] O. Holland, E. Steinbach, R. V. Prasad, Q. Liu, Z. Dawy, A. Aijaz, N. Pappas, K. Chandra, V. S. Rao, S. Oteafy, M. Eid, M. Luden, A. Bhardwaj, X. Liu, J. Sachs, and J. Araújo. The IEEE 1918.1 Tactile Internet Standards Working Group and its Standards. *Proceedings of the IEEE*, 107(2):256–279, February 2019.

[6] E. Steinbach, M. Strese, M. Eid, X. Liu, A. Bhardwaj, Q. Liu, M. Al-Jaafreh, T. Mahmoodi, R. Hassen, A. El Saddik, and O. Holland. Haptic Codecs for the Tactile Internet. *Proceedings of the IEEE*, 107(2):447–470, February 2019.

[7] M. Grieves and J. Vickers. *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*, pages 85–113. Springer International Publishing, Cham, 2017.

[8] T. L. Ghezzi and O. C. Corleta. 30 Years of Robotic Surgery. *World Journal of Surgery*, 40(10):2550–2557, October 2016.

[9] J. Marescaux, J. Leroy, F. Rubino, M. Smith, M. Vix, M. Simone, and D. Mutter. Transcontinental Robot-Assisted Remote Telesurgery: Feasibility and Potential Applications. *Annals of Surgery*, 235(4):487–492, April 2002.

[10] A. Aijaz, Z. Dawy, N. Pappas, M. Simsek, S. Oteafy, and O. Holland. Toward a tactile Internet reference architecture: Vision and progress of the IEEE P1918. 1 standard. *arXiv preprint arXiv:1807.11915*, 2018.

[11] ITU-R. Minimum requirements related to technical performance for IMT-2020 radio interface(s). (Rep. No. M.2410-0), November 2017.

[12] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349.

[13] S. Deering and R. Hinden. RFC 8200: Internet Protocol, Version 6 (IPv6) Specification. *IETF*, July 2017.

[14] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach*. Pearson, Boston, MA, 7 edition, 2016.

[15] Number of internet of things (iot) connected devices worldwide in 2018, 2025 and 2030, 2018. `https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/`.

[16] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930 (Best Current Practice), March 1996.

[17] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[18] M. T. Beck, M. Werner, S. Feld, and S. Schimper. Mobile edge computing: A taxonomy. In *Proc. of the Sixth International Conference on Advances in Future Internet*, pages 48–55. Citeseer, 2014.

[19] Samsung 6G Vision. *Whitepaper*, July 2020.

[20] M. Latva-aho and K. Leppänen. 6G Research Visions 1. Key Drivers and Research Challenges for 6G Ubiquitous Wireless Intelligence. resreport 9789526223544, University of Oulu, Finland, September 2019.

[21] P. Yang, Y. Xiao, M. Xiao, and S. Li. 6G Wireless communications: Vision and potential techniques. *IEEE Network*, 33(4):70–75, 2019.

[22] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson. 6g wireless systems: Vision, requirements, challenges, insights, and opportunities. *Proceedings of the IEEE*, 109(7):1166–1199, 2021.

[23] E. Steinbach, S. Hirche, M. Ernst, F. Brandi, R. Chaudhari, J. Kammerl, and I. Vittorias. Haptic communications. *Proceedings of the IEEE*, 100(4):937–956, April 2012.

[24] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh. Realizing the Tactile Internet: Haptic Communications over Next Generation 5G Cellular Networks. *IEEE Wireless Communications*, 24(2):82–89, April 2017.

[25] E. Steinbach, S. Hirche, J. Kammerl, I. Vittorias, and R. Chaudhari. Haptic Data Compression and Communication. *IEEE Signal Processing Magazine*, 28(1):87–96, January 2011.

[26] T. B. Sheridan and W. L. Verplank. Human and computer control of undersea teleoperators. techreport, Massachusetts Inst of Technology, Cambridge, MA, July 1978.

[27] M. Dohler, T. Mahmoodi, M. A. Lema, M. Condoluci, F. Sardis, K. Antonakoglou, and H. Aghvami. Internet of skills, where robotics meets AI, 5G and the Tactile Internet. In *2017 European Conference on Networks and Communications (EuCNC)*. IEEE, June 2017.

[28] R. Behrens and G. Pliske. Human-Robot Collaboration: Partial Supplementary Examination [of Pain Thresholds] for Their Suitability for Inclusion in Publications of the DGUV and Standardization. Research report, Fraunhofer IFF, Sandtorstr. 22, 39106 Magdeburg, Germany, December 2019.

[29] X. Zhang, Q. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng. Networked control systems: a survey of trends and techniques. *IEEE/CAA Journal of Automatica Sinica*, 7(1):1–17, January 2020.

[30] F. Mager, D. Baumann, R. Jacob, L. Thiele, S. Trimpe, and M. Zimmerling. Feedback control goes wireless: guaranteed stability over low-power multi-hop networks. *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems - ICCPS*, February 2019.

[31] A. I. Maass, D. Nei, R. Postoyan, and P. M. Dower. Lp stability of networked control systems implemented on WirelessHART. *Automatica*, 109:108514, November 2019.

[32] E. Steinbach. Teleoperation over 5G networks. Presentation Slides. Symposium: 5G Stärken, Schwächen und Chancen in der Medizin; April 13, 2018; Klinikum rechts der Isar, TU München, Ismaninger Str. 22, 81675 München.

[33] K. Nichols, S. L. Blake, F. Baker, and D. L. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. *IETF RFC 2474*, (2474), December 1998.

[34] J. Wroclawski. The Use of RSVP with IETF Integrated Services. *IETF RFC 2210*, September 1997. `http://www.rfc-editor.org/rfc/rfc2210.txt`.

[35] J. Wroclawski. Specification of the Controlled-Load Network Element Service. *IETF RFC 2211*, September 1997. `http://www.rfc-editor.org/rfc/rfc2211.txt`.

[36] M. Meehan, S. Razzaque, M. C. Whitton, and F. P. Brooks. Effect of Latency on Presence in Stressful Virtual Environments. *Proceedings of IEEE Virtual Reality*, pages 141–148, March 2003.

[37] P. DiZio and J. R. Lackner. Circumventing side effects of immersive virtual environments. *Advances in human factors/ergonomics*, pages 893–896, 1997.

[38] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach. Model-Mediated Teleoperation: Toward Stable and Transparent Teleoperation Systems. *IEEE Access*, 4:425–449, January 2016.

[39] Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pages 1–4379, 2021.

[40] T. B. Sheridan. Space Teleoperation Through Time Delay: Review and Prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, November 1993.

[41] C. Bachhuber, E. Steinbach, M. Freundl, and M. Reisslein. On the Minimization of Glass-to-Glass and Glass-to-Algorithm Delay in Video Communication. *IEEE Transactions on Multimedia*, 20(1):238–252, January 2018.

[42] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van. The tactile internet: vision, recent progress, and open challenges. *IEEE Communications Magazine*, 54(5):138–145, 2016.

[43] A. Aijaz and M. Sooriyabandara. The Tactile Internet for Industries: A Review. *Proceedings of the IEEE*, 107(2):414–435, 2019.

[44] J. C. Laprie. *Dependability: Basic Concepts and Terminology.* Springer, 1992.

[45] T. HöSSler, M. Simsek, and G. P. Fettweis. Mission Reliability for URLLC in Wireless Networks. *IEEE Communications Letters*, 22(11):2350–2353, November 2018.

[46] J. M. Thompson, M. P. Ottensmeyer, and T. B. Sheridan. Human factors in telesurgery: effects of time delay and asynchrony in video and control feedback with local manipulative assistance. *Telemedicine Journal*, 5(2):129–137, 1999.

[47] H. Mayer, I. Nagy, A. Knoll, E. U. Braun, R. Bauernschmitt, and R. Lange. Haptic Feedback in a Telepresence System for Endoscopic Heart Surgery. *Presence: Teleoperators and Virtual Environments*, 16(5):459–470, October 2007.

[48] B. Holfeld, D. Wieruch, T. Wirth, L. Thiele, S. A. Ashraf, J. Huschke, I. Aktas, and J. Ansari. Wireless communication for factory automation: An opportunity for LTE and 5G systems. *IEEE Communications Magazine*, 54(6):36–43, June 2016.

[49] V. Vukadinovic, K. Bakowski, P. Marsch, I. D. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. Lister, and I. Thibault. 3GPP C-V2X and IEEE 802.11 p for Vehicle-to-Vehicle communications in highway platooning scenarios. *Ad Hoc Networks*, 74:17–29, 2018.

[50] European Telecommunications Standards Institute. 5G; Service requirements for next generation new services and markets. Technical Specification (TS) 22.261, European Telecommunications Standards Institute, October 2019. Version 15.8.0.

[51] X. Jiang, H. Shokri-Ghadikolaei, G. Fodor, E. Modiano, Z. Pang, M. Zorzi, and C. Fischione. Low-Latency Networking: Where Latency Lurks and How to Tame It. *Proceedings of the IEEE*, 107(2):280–306, February 2019.

[52] N. Promwongsa, A. Ebrahimzadeh, D. Naboulsi, S. Kianpisheh, F. Belqasmi, R. Glitho, N. Crespi, and O. Alfandi. A Comprehensive Survey of the Tactile Internet: State-of-the-art and Research Directions. *IEEE Communications Surveys & Tutorials*, September 2020.

[53] S. K. Sharma, I. Woungang, A. Anpalagan, and S. Chatzinotas. Toward Tactile Internet in Beyond 5G Era: Recent Advances, Current Issues, and Future Directions. *IEEE Access*, 8:56948–56991, March 2020.

[54] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, M.-N. Garcia, T. Hossfeld, S. Jumisko-Pyykkö, C. Keimel, M.-C. Larabi, et al. Qualinet white paper on definitions of quality of experience. 2013.

[55] F. F. Reichheld. The one number you need to grow. *Harvard business review*, 81(12):46–55, December 2003.

[56] R. Chaudhari, E. Steinbach, and S. Hirche. Towards an objective quality evaluation framework for haptic data reduction. *Proceedings of the IEEE World Haptics Conference*, pages 539–544, June 2011.

[57] R. Hassen and E. Steinbach. HSSIM: An Objective Haptic Quality Assessment Measure for Force-Feedback Signals. *Proceedings of the 10th International Conference on Quality of Multimedia Experience (QoMEX)*, May 2018.

[58] J. P. Verburg, H. J. C. Kroep, V. Gokhale, R. V. Prasad, and V. Rao. Setting the Yardstick: A Quantitative Metric for Effectively Measuring Tactile Internet. *Proceedings of the IEEE Conference on Computer Communications*, pages 1937–1946, July 2020.

[59] K. Polachan, P. T.V, C. Singh, and D. Panchapakesan. Quality of Control Assessment for Tactile Cyber-Physical Systems. *Proceedings of the 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2019.

[60] A. Hamam, A. E. Saddik, and J. Alja'Am. A Quality of Experience Model for Haptic Virtual Environments. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 10(3):1–23, April 2014.

[61] S. Blumenthal, N. Hochgeschwender, E. Prassler, H. Voos, and H. Bruyninckx. An approach for a distributed world model with QoS-based perception algorithm adaptation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1806–1811, September 2015.

[62] J. Schiller. *Mobile Communications.* Addison-Wesley, 2nd edition edition, 2003.

[63] M. Günes. On the scientific value of large-scale testbeds for wireless multi-hop networks, 2017.

[64] K. Wehrle, M. Günes, and J. Gross. *Modeling and tools for network simulation.* Springer Science & Business Media, 2010.

[65] OMNet++ Discrete Event Simulator. `https://omnetpp.org/`, accessed Jun 2022.

[66] MIoT-Lab. `http://comsys.ovgu.de/miot_lab`.

[67] K. Kientopf, M. Buschsieweke, and M. Günes. Technical report: Designing a testbed for wireless communication research on embedded devices. In *18. GI/ITG KuVS Fachgespräch Sensornetze – FGSN 2019*, pages 41 – 44, Magdeburg, September 2019.

[68] E. Baccelli, C. Gündogan, O. Hahm, P. Kietzmann, M. S. Lenders, H. Petersen, K. Schleiser, T. C. Schmidt, and M. Wählisch. RIOT: An Open Source Operating System for Low-End Embedded Devices in the IoT. *IEEE Internet of Things Journal*, 5(6):4428–4440, 2018.

[69] RIOT OS. `http://riot-os.org`.

[70] M. Günes, F. Juraschek, B. Blywis, and O. Watteroth. DES-CRIPT - A Domain Specific Language for Network Experiment Descriptions. In *Inproceedings of the International Conference on Next Generation Wireless Systems (NGWS) 2009*, Melbourne, Australia, 12-14, October 2009.

[71] INET Framework for OMNet++. `https://inet.omnetpp.org/`, accessed Jun 2022.

[72] Simu5G simulator for 5G New Radio networks for OMNet++. `http://http://simu5g.org/`, accessed Jun 2022.

[73] M. Grieves. Digital twin: Manufacturing excellence through virtual factory replication. White paper, Digital Twin Institute, March 2015.

[74] A. A. Ateya, A. Muthanna, A. Vybornova, I. Gudkova, Y. Gaidamaka, A. Abuarqoub, A. D. Algarni, and A. Koucheryavy. Model Mediation to Overcome Light LimitationsToward a Secure Tactile Internet System. *Journal of Sensor and Actuator Networks*, 8(1), January 2019.

[75] Q. Bi. Ten trends in the cellular industry and an outlook on 6g. *IEEE Communications Magazine*, 57(12):31–36, 2019.

[76] H. Fourati, R. Maaloul, L. Chaari, and M. Jmaiel. Comprehensive survey on self-organizing cellular network approaches applied to 5g networks. *Computer Networks*, 199:108435, 2021.

[77] E. Dahlman. *5G NR: the next generation wireless access technology.* Academic Press, London, United Kingdom, 2018. Erik Dahlman, Stefan Parkvall, Johan Sköld.

[78] M. Fritzsche, N. Elkmann, and E. Schulenburg. Tactile sensing: A key technology for safe physical human robot interaction. In *Proceedings of the 6th International Conference on Human-robot Interaction*, pages 139–140, 2011.

[79] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury. Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research. *IEEE Communications Surveys Tutorials*, 21(1):88–145, Firstquarter 2019.

[80] IEEE Std 802.1CB-2017: Local and Metropolitan Area Networks - Frame Replication and Elimination for Reliability. *IEEE Standard*, May 2017.

[81] IEEE Std 802.1CM-2018: Local and Metropolitan Area Networks - Time Sensitive Networking for Fronthaul. *IEEE Standard*, May 2018.

[82] IEEE Std 802.1C: Local and Metropolitan Area Networks - Bridges and Bridged Networks. *IEEE Standard*, May 2018.

[83] IETF Detnet Working Group.

[84] B. Varga, J. Farkas, L. Berger, D. Fedyk, A. G. Malis, S. Bryant, and J. Korhonen. DetNet Data Plane: IP over MPLS. *IETF Draft*, November 2019. draft-ietf-detnet-ip-over-mpls.

[85] N. Finn and P. Thubert. Deterministic Networking Problem Statement. RFC 8557, May 2019.

[86] E. Grossman. Deterministic Networking Use Cases. RFC 8578, May 2019.

[87] B. Varga, J. Farkas, L. Berger, D. Fedyk, A. G. Malis, S. Bryant, and J. Korhonen. DetNet Data Plane Framework. *IETF Draft*, October 2019. draft-ietf-detnet-data-plane-framework.

[88] N. Finn, P. Thubert, B. Varga, and J. Farkas. Deterministic Networking Architecture. RFC 8655, October 2019.

[89] D. O. Awduche, L. Berger, D.-H. Gan, T. Li, D. V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209, December 2001.

[90] ETSI Work Item DGR/IP6-0014; IPv6 based Tactile Internet.

[91] ETSI GR NGP 010 V1.1.1: Next Generation Protocols (NGP); Recommendation for New Transport Technologies. Etsi group report, September 2018.

[92] X. Xu, B. Cizmeci, A. Al-Nuaimi, and E. Steinbach. Point Cloud-based Model-mediated Teleoperation with Dynamic and Perception-based Model Updating. *IEEE Transactions on Instrumentation and Measurement*, 63(11):2558 – 2569, 2014.

[93] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski. Modeling internet backbone traffic at the flow level. *IEEE Transactions on Signal Processing*, 51(8):2111–2124, July 2003.

[94] D. Bertsekas and J. Tsitsiklis. *Introduction to Probability.* Athena Scientific, Belmont, Massachusetts, second edition edition, 2008.

[95] M. Jaber, F. J. Lopez-Martinez, M. A. Imran, A. Sutton, A. Tukmanov, and R. Tafazolli. Wireless backhaul: Performance modeling and impact on user association for 5g. *IEEE Transactions on Wireless Communications*, 17(5):3095–3110, 2018.

[96] G. Zhang, T. Q. S. Quek, A. Huang, M. Kountouris, and H. Shan. Delay modeling for heterogeneous backhaul technologies. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 1–6, 2015.

[97] R. Jain. *The Art of Computer Systems Performance Analysis*. Wiley, 1992.

[98] L. Lipsky. *Queueing Theory*. Springer, New York, NY, 2009.

[99] J. D. C. Little. A proof for the queuing formula: L= w. *Operations Research*, 9(3):383–387, 1961.

[100] M. Stojanova, T. Begin, and A. Busson. Conflict graph-based model for IEEE 802.11 networks: A Divide-and-Conquer approach. *Performance Evaluation*, 130:64 – 85, 2019.

[101] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.

[102] S. Frohn, S. Gübner, and C. Lindemann. Analyzing the effective throughput in multi-hop IEEE 802.11n networks. *Computer Communications*, 34(16):1912 – 1921, 2011.

[103] H. Wu, Y. Lin, S. Cheng, Y. Peng, and K. Long. IEEE 802.11 distributed coordination function: Enhancement and analysis. *Journal of Computer Science and Technology*, 18(5):607–614, 2003.

[104] F. Calì, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transactions on networking*, 8(6):785–799, 2000.

[105] P. Chatzimisios, A. C. Boucouvalas, and V. Vitsas. Performance analysis of the IEEE 802.11 MAC protocol for wireless LANs. *International Journal of Communication Systems*, 18(6):545–569, 2005.

[106] P. Chatzimisios, V. Vitsas, A. C. Boucouvalas, and M. Tsoulfa. Achieving performance enhancement in IEEE 802.11 WLANs by using the DIDD backoff mechanism. *International Journal of communication systems*, 20(1):23–41, 2007.

[107] M. Maadani and S. A. Motamedi. A comprehensive DCF performance analysis in noisy industrial wireless networks. *International Journal of Communication Systems*, 29(11):1720–1739, 2016.

[108] T. Sakurai and H. L. Vu. MAC access delay of IEEE 802.11 DCF. *IEEE Transactions on Wireless Communications*, 6(5):1702–1710, 2007.

[109] J. S. Vardakas, M. K. Sidiropoulos, and M. D. Logothetis. Performance behaviour of IEEE 802.11 distributed coordination function. *IET circuits, devices & systems*, 2(1):50–59, 2008.

[110] P. Raptis, V. Vitsas, and K. Paparrizos. Packet delay metrics for IEEE 802.11 distributed coordination function. *Mobile Networks and Applications*, 14(6):772–781, 2009.

[111] G. Tian and Y.-C. Tian. Modelling and performance evaluation of the IEEE 802.11 DCF for real-time control. *Computer Networks*, 56(1):435 – 447, 2012.

[112] A. Zocca, S. Borst, J. van Leeuwaarden, and F. Nardi. Delay performance in random-access grid networks. *Performance Evaluation*, 70(10):900 – 915, 2013. Proceedings of IFIP Performance 2013 Conference.

[113] T. Kanematsu, K. Sanada, Z. Li, T. Pei, Y.-J. Choi, K. Nguyen, and H. Sekiya. Throughput and delay analysis for ieee 802.11 multi-hop networks considering data rate. *International Journal of Distributed Sensor Networks*, 16(9), September 2020.

[114] F. Engelhardt, C. Rong, and M. Günes. Towards Tactile Wireless Multi-Hop Networks - The Tactile Coordination Function as EDCA Supplement. In *2019 Wireless Telecommunications Symposium (WTS 2019)*, New York City, USA, April 2019.

[115] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 28–28. USENIX Association, 2005.

[116] D. Bertsimas and D. Nakazato. The distributional little's law and its applications. *Operations Research*, 43(2):298–310, April 1995.

[117] M. Maadani and S. A. Motamedi. A simple and closed-form access delay model for reliable IEEE 802.11-based wireless industrial networks. *Wireless personal communications*, 75(4):2243–2268, 2014.

[118] M. F. Neuts. Renewal processes of phase type. *Naval Research Logistics Quarterly*, 25(3):445–454, September 1978.

[119] Ieee standard for information technology–local and metropolitan area networks–specific requirements–part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications - amendment 8: Medium access control (mac) quality of service enhancements. *IEEE Std 802.11e-2005*, 2005.

[120] M. Günes, M. Hecker, and I. Bouazizi. Influence of adaptive rts/cts retransmissions on tcp in wireless and ad-hoc networks. In *Proceedings of the 8th IEEE Symposium on Computer and Communications, (ISCC 2003)*, volume II, pages 855–860, Antalya, Turkey, June, July 2003. IEEE Computer Society.

[121] M. Günes and D. Vlahovic. The Performance of the TCP/RCWE Enhancement for Ad-Hoc Networks. In M. D. Antonio Corradi, editor, *Proceedings of the Seventh IEEE Symposium on Computers and Communications, ISCC 2002*, pages 43–48. IEEE, July 2002.

[122] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[123] Gazebo. `http://gazebosim.org/`, accessed Jun 2022.

[124] CoppeliaSim. `http://www.coppeliarobotics.com/`, accessed Jun 2022.

[125] M. Günes, B. Blywis, F. Juraschek, and O. Watteroth. Experimentation Made Easy. In ICST, editor, *Proceedings of the First International Conference on Ad Hoc Networks*, volume 28 of *28*, pages 493–505, Ontario, Canada, September 2009. Springer Berlin Heidelberg.

[126] M. Günes, F. Juraschek, B. Blywis, Q. Mushtaq, and J. Schiller. A Testbed for Next Generation Wireless Networks Research. *Special Issue PIK on Mobile Ad-hoc Networks*, 32(4):208–212, January 2009.

[127] M. Günes, F. Juraschek, and B. Blywis. An Experiment Description Language for Wireless Network Research. *Journal of Internet Technology (JIT), Special Issue for Mobile Internet*, 11(4):465–471, 7 2010.

[128] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). *IETF RFC 7252*, June 2014.

[129] B. Blywis, M. Günes, F. Juraschek, P. Schmidt, and P. Kumar. DES-SERT: A framework for structured routing protocol implementation. In *IFIP Wireless Days conference (WD'09)*, pages 1–6, Paris, France, Dec 2009. IEEE.

[130] Robot Operating System (ROS). `https://www.ros.org/`, accessed Jun 2022.

[131] F. Engelhardt, S. Herbrechtsmeyer, and M. Günes. Kinesthetic Coding Based on the Fast Wavelet Transform for Remote-Controlling a Quadrotor Drone. In *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, pages 157–162, 2022.

[132] A. Goldsmith. *Wireless Communications.* Cambridge University Press, 2005.

Appendix

Characteristics of Probability Distributions

In Section 2.3 we discussed several methods for tracking measurement data, for example latency, throughput and packet loss. Measurement data is based on random processes that we intend to observe through our measurements, and measurements themselves are momentary images of these processes, and thus samples drawn from these random variables. We summarize here characteristics from random variables that can serve as single-number expressions of the underlying random processes.

Table A.1: Characteristics for probability distributions.

Characteristics for continuous random variables X and Y with probability density functions $f_X(x)$ and $f_Y(y)$, and samples $\{x_1, ..., x_i, ..., x_n\}$ and $\{y_1, ..., y_j, ..., y_m\}$ drawn from X and Y, respectively. Let the relative frequency of a sample specimen x_i be p_{x_i}.

Name	**Symbol**	**Calculation**	**Application**
Characteristics defined on probability distributions			
Expected value	$\mathrm{E}[X]$	$\int_{-\infty}^{\infty} x f_X(x)\,\mathrm{d}x$	Averaging
Variance	$\mathrm{Var}(X)$	$\mathrm{E}\left[(\mathrm{E}[X] - X)^2\right]$	Error estimation
Standard deviation	σ_X	$\sqrt{\mathrm{Var}(X)}$	Error estimation
Covariance	$\mathrm{Cov}(X, Y)$	$\mathrm{E}[XY] - \mathrm{E}[X]\,\mathrm{E}[Y]$	Correlation
Correlation coefficient	$\rho\{X, Y\}$	$\frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$	Correlation
$100p$-Percentile	X_p	$F_X^{-1}(p)$	Error estimation
Characteristics defined on samples of random variables			
Sample mean	$\overline{x}$	$\sum_{i=1}^{n} p_{x_i} x_i$	Averaging
Geometric mean	$\dot{x}$	$\sqrt[n]{\prod_{i=1}^{n} x_i}$	Averaging
Harmonic mean	$\ddot{x}$	$\frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$	Averaging
Sample median	–	$x_{\lfloor n/2 \rfloor}$	Averaging
Sample mode	–	$\underset{p_{x_i}}{\mathrm{argmax}}\,\{x_i\}$	Averaging
Sample variance	s^2	$\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2$	Error estimation
Sample standard deviation	s	$\sqrt{s^2}$	Error estimation
Range	–	$\min x_i, \max x_i$	Error estimation

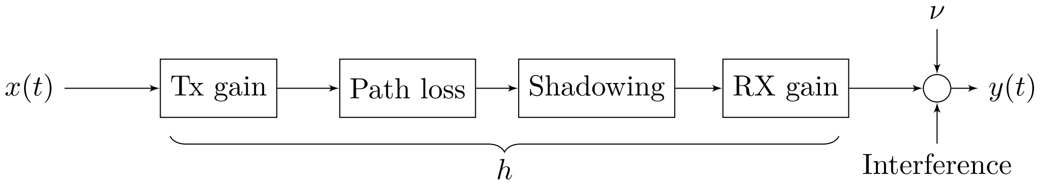


Figure A.1: Channel model.

Wireless Signal Propagation

For an understanding of wireless transmission systems, it is necessary to view at basic signal propagation at the physical layer. Radio transmissions consist of a signal $x(t)$ in the electromagnetic spectrum. This signal is generated at the transmitter, where it is modulated with information. The receiver, if being able to receive the full, unattenuated and undistorted signal $x(t)$, will be able to decode ts information without any losses.

Channel Model

The wireless channel is not lossless. It imposes the original transmitted signal with additive and multiplicative components ν and h, respectively:

$$y(t) = hx(t) + \nu \tag{A.1}$$

Here, $y(t)$ is the signal captured at the receiver. h and ν are complex random variables. h is the channel characteristic, representing signal attenuation and phase shift. $\nu \sim N(0, \sigma)$ is an additive white gaussian noise term.

Signals from several transmitters behave additive, which means that interference from other transmitters cannot be distinguished from one another. Suppose a signal $x_0(t)$ is emitted by a transmitter to a specific receiver, and at the same time, n competing transmitters emit signals $x_i(t), 1 \leq i \leq n$. Then the signals $x_i(t)$ appear as *interference*, since the additive nature of the channel does not allow the receiver to distinguish between each of the signals.

$$y(t) = \underbrace{h_0 x_0(t)}_{\text{signal of interest}} + \underbrace{\sum_{i=1}^{n} h_i x_i(t)}_{\text{interference term}} + \underbrace{\nu}_{\text{noise term}}$$

Figure A.1 summarizes the channel model. Note that the channel characteristics h_i appear as different constants for each of the transmitters, and in general depend on the specific conditions between the respective receiver-transmitter pair.

Signal Attenuation

The complex constant h in Equation A.1 is responsible for reduction of signal amplitude and thus signal power. The imaginary part of h can also represent a phase shift. We consider here only relatively long-term phenomena; short-term channel effects with bandwidths comparable to the signal bandwidth have to be modeled by a time-variant function $h(t)$ [132]. Attenuation effects can be distinguished by *path loss* effects, *shadowing*, and *fading*. Path loss occurs due to the energy dissipation into the free space, with field lines of the electromagnetic field becoming less dense with increasing distance from the antenna. Shadowing refers to attenuation caused by obstacles through absorption, reflection, scattering, and diffraction, and can be considered relatively time-invariant. Fading, on the contrary, is a time-variant, stochastic effect that reduces signal strength due to multi-path transmission.

Attenuation, in general, is defined as the relation of received signal power to transmitted signal power. The signal power p_x of a signal is defined as the mean value of the squared waveform $x(t)$:

$$p_x = \int_{t=0}^{\infty} |x(t)|^2 \, \mathrm{d}t.$$

The power is often denoted in logarithmic form in dBm, where it is normalized to a value of 1 Milliwatt:

$$p_{x,\mathrm{dBm}} = 10 \times \log_{10} \frac{p_x}{1\,\mathrm{mW}}.$$

Path Loss

A common model is the free space path loss model that covers Line-of-Sight (LOS) transmission. It is defined over the attenuation of the signal power assuming a non-obstructed signal propagation over a distance d.

Then the free space path loss is defined for the the received signal power $p_{y,\mathrm{dBm}}$ as

$$p_y = \left(\frac{\lambda}{4\pi r}\right)^2 p_x.$$

$$p_{y,\mathrm{dBm}} = 20 \log_{10} \left(\frac{4\pi d}{\lambda}\right).$$

Channel Capacity

Assuming that the signal $x_0(t)$ is a payload signal, and $x_i(t)$ are the interference signals, then the Signal to Noise and Interference Ratio (SNIR) at the receiver is defined as the fraction

$$\gamma = \frac{p(h_0 x_0(t))}{p(\nu + \sum_{i=1}^{n} h_i x_i(t))} = \frac{|h_0|^2 p(x_0(t))}{p(\nu) + \sum_{i=1}^{n} |h_i|^2 p(x_i(t))}$$

Probability Theory

Probability theory is required in statistical analysis to assess the likelihood of some event to be true (or false). We repeat basic elements here to define the notation within this thesis.

The probability of an event A is denoted as $\Pr\{A\}$. The notion $\Pr\{A\} = 0.2$ means that A occurs with 20 % probability. A random variable X, in contrast, is a numerical value (either in $\mathbb{N}$ for discrete or in $\mathbb{R}$ for continuous random variables), that takes any value within a specific range as defined by a *probability distribution*. The notion $\Pr\{X < 0.2\} = 0.3$ means that X takes a value of less than 0.2 with a probability of 30 %. The same relation can be expressed with the term $F_X(0.2) = 0.3$, where the function $F_X(\cdot)$ is called the *CDF* of the random variable X. Its derivative $\frac{\mathrm{d}}{\mathrm{d}x} F_X(x) = f_X(x) = \Pr\{X = x\}$, in case of a continuous random variable, is called the *PDF* of X.

Probabilities of an event A or a random variable X can be given conditional to some other event B or random variable Y. Conditional probabilities can also be expressed by conditional PDFs:

$$\Pr\{A|B\} = \frac{\Pr\{A \cap B\}}{\Pr\{B\}}$$

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

The event $A \cap B$ is a compound event that indicates that both A and B occur. The function $f_{X,Y}(x,y)$ is the respective form of the joint probability density in PDF notation. Two variables X and Y are called *independent*, iff $f_{X|Y}(x|y) = f_X(x)$ for all x, and thus $f_{X;Y}(x;y) = f_X(x) \times f_Y(y)$.

We further denote the *expected value* and the *variance* of a random variable X as $\mathrm{E}\left[X\right]$ and $\mathrm{Var}\left(X\right)$, respectively. The *covariance* between X and Y is denoted as $\mathrm{Cov}\left(X, Y\right)$. Some more notation is shown in Appendix 12.0.1.

Markov Modeling

Markov chains can often be used to model computer systems and are themselves the basis for queueing systems analysis.

A Markov chain is defined by a set of states $\mathcal{S} = \{1, ..., n\}$, a matrix of transition probabilities $p_{i;j}$ between the states $i, j \in \mathcal{S}$, and a vector σ_i of initial probabilities for the states $i \in \mathcal{S}$.

A *(discrete) Markov chain* X is an infinite sequence $X_0, X_1, X_2, ...$ of random variables that each take values of $\mathcal{S}$ that satisfy the *Markov property*

$$\begin{aligned} &\Pr\left\{X_{n+1} = j | X_n = i \cap X_{n-1} = i_{n-1}, \cap X_{n-2} = i_{n-2} \cap ... \cap X_0 = i_0\right\} \\ &\quad = \Pr\left\{X_{n+1} = j | X_n = i\right\} \\ &\quad = p_{i;j}. \end{aligned}$$

It is also called the memorylessness property, since a new state is only dependent on the last state, but not on the complete history of states.

When we analyze Markov chains, we are often interested in either the transient behavior, or its steady-state behavior. The steady-state behavior is characterized as a stable probability distribution $\pi(i)$ that emerges as a long-term behavior of the network, i.e.,

$$\pi(i) = \lim_{n \to \infty} \Pr\left\{X_n = i\right\}.$$

The transient behavior, in contrast, characterizes the behavior of the system until that point, e.g., how many steps are required to reach the steady-state.

In this thesis, we will use discrete Markov processes as a basis for analyzing queueing behavior in Chapter 8.

Modeling of Queueing Systems

A queueing model is a means of statistical analysis for computer systems, where events occur from a given event pool (either finite or infinite) and have to be processed by a *server*. Events arrive at random time points and enter the server directly if it is not occupied by another event yet. If so it enters a buffer first, where it is queued together with other events to wait for its processing by the server. The event arrivals, as well as the processing time needed by the server, are subject to random distributions, called *arrival process* or *service process*, respectively. Servers can also be duplicated for parallel processing. They share the same arrival process and the same queue.

Queueing systems are classified according to the Kendall notation $X/\,Y/\,N$, where X describes the arrival process, Y the service process, and N the number of duplicated servers that work in parallel. Common distributions for X and Y are the *Poisson* process (also called Markov process), denoted as M, the *deterministic* process denoted as D, and the *general* process, denoted as G. The Poisson process M is the most often used one. Here, the arrival (or service) process obeys exponential distribution, which means that the inter-arrival time λ (or, inter-departure time μ, respectively) is exponentially distributed. The process is, therefore, defined by the arrival rate λ or the service rate λ, which are the parameters of the respective exponential distributions. A queue with a single server and both arrival and service processes being Poisson processes is denoted a $M/\,M/\,1$ queue with arrival rate λ and service rate μ. It is often denoted by the symbol in Figure 8.9.

A *queuing discipline* (sometimes also called *scheduling discipline*) is responsible to pick the next event from the queue as soon as the server is done processing an event. In this thesis,

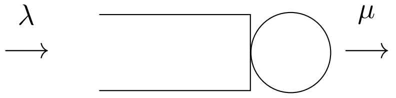


Figure A.2: Basic notation of a *M/ M/ 1* queueing model with arrival rate λ and service rate μ.

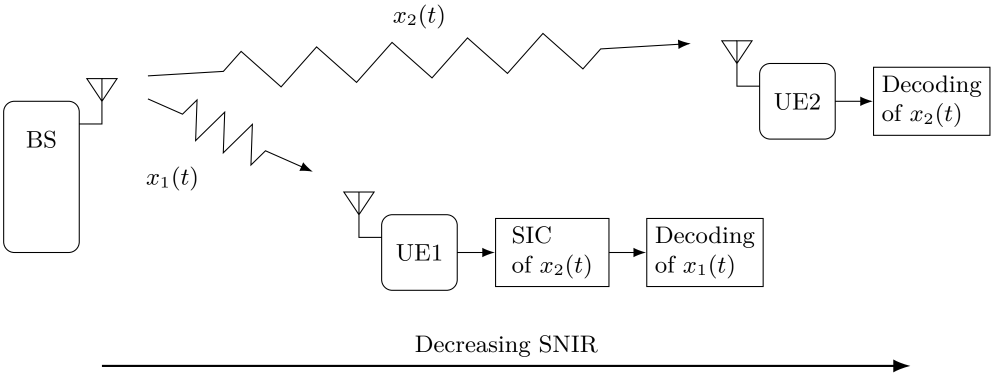


Figure A.3: Principle of Non-orthogonal Multiple Access (NOMA), adapted from [1].

we always assume a priority-based first-in first-out (FIFO) queueing discipline, where events are assigned a priority. Highest priority events are processed first, but within events of the same priority, the first event that entered the queue will be processed first.

Queueing systems with general distributions can be modeled, however, they can only de described and analyzed with a very limited set of laws, e.g., Little's law, that we discuss in more detail in Chapter 6. In contrast, queueing systems with simple process descriptions, as the *M/ M/ 1* queue, are often too undercomplex to model wireless communication systems. A powerful tool for both modeling and analyzing complex queueing systems is the Matrix-Exponential (*ME*) distribution [98], which we discuss in detail in Chapter 8.

Calculation of Channel Capacity

In this section, we review the calculation of the channel capacity for two transmission techniques, NOMA transmission and MIMO transmission.

Non-orthogonal Multiple Access (NOMA) transmission

NOMA is a relatively recent technique in which the channel resources are not divided orthogonally between users. It can increase spectral efficiency, especially in situations where the channel is asynchronous [1]. In asynchronous channels, multiple users have different attenuation characteristics, such that for "strong" users, the channel attenuation towards the base station is small in comparison to "weak" users. With power-level NOMA, strong users are granted relatively low-power transmissions, such that more channel resources can be transferred to weak users. Weak users are then able to decode signals addressed to them by treating signals dedicated to strong users as background interference. Strong users, on the other hand, can decode signals addressed to them by a technique called successive interference cancellation (SIC). Since for strong users, those signals can also be decoded directly, they can re-code and subtract them from the received signal, effectively cancelling them. Figure A.3 demonstrates this operating principle where a strong user (UE1) can use SIC to cancel the stronger signal that is sent to UE2, thus filtering out the stronger signal $x_2(t)$. This allows the base station to serve UE1 and UE2 simultaneously. With this technique, resource allocation can exceed 100 % compared to classic orthogonal multiple access.

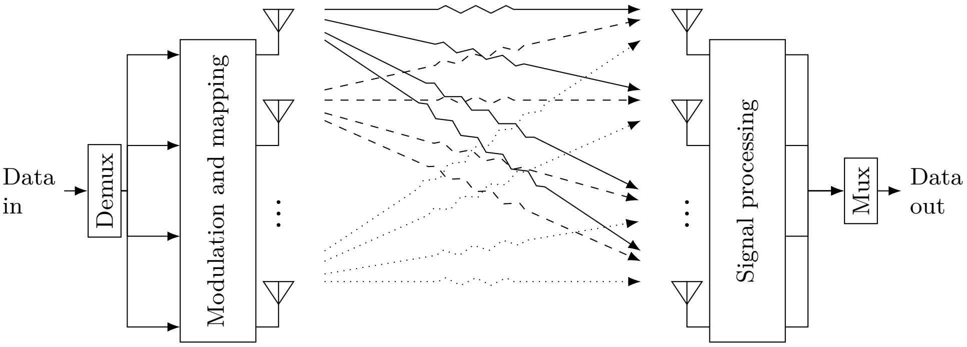


Figure A.4: Principle of Multiple Input/Multiple Output (MIMO) spatial multiplexing, adapted from [2].

The overall channel capacity of such systems is given by [1]

$$C = \sum_{i=1}^{K} C_i$$

with $1 \leq i \leq K$. The individual capacities C_i for each user differ in the uplink and the downlink. The channel capacity for individual users is in the downlink

$$C_i = B \times \log_2 \left(1 + \frac{p_i |h_i|^2}{N + |h_i|^2 \sum_{j=1}^{i-1} p_j} \right)$$

and in the uplink

$$C_i = B \times \log_2 \left(1 + \frac{p_i |h_i|^2}{N + \sum_{j=1}^{i-1} |h_j|^2 p_j} \right).$$

N is the Gaussian noise and the inter-cell interference, p_i is the transmission power of user i, and h_i is the coefficient describing the channel characteristic between user i and the base station. Without loss of generality, it is defined that $h_i \geq h_2 \geq ... \geq h_K$, i.e. user 1 is the "strongest" and user K the "weakest". This leads to a respective inverted power allocation $p_1 \leq p_2 \leq ... \leq p_k$.

It can be seen in both equations that the overall channel capacity is bigger than with classic orthogonal multiple access, since the interference terms in the denominators ignore parts of the respective weaker users.

Multiple Input/Multiple Output (MIMO) Transmission Systems

MIMO is a means of increasing channel capacity with antenna arrays [2]. The diversity of the individual channels from antenna to antenna results in a potential increase in channel capacity. Figure A.4 shows the mode of operation.

For MIMO systems, the channel capacity formula is given in a vector form. Assuming M transmitting antennae and N receiving antennae, it is given by

$$C = B \times \log_2 \det \left(I_M + \frac{\rho}{N} H H^* \right) \text{ [2].}$$

Here, $H = [h_{ij}]_{M \times N}$ is the channel matrix, containing channel coefficients. h_{ij} represents the channel coefficient between transmitter antenna i and receiver antenna j. H^* denotes the transpose conjugate of H.

Own Publications

- Frank Engelhardt and Mesut Günes. A /sys filesystem for the internet of things. In *2022 IEEE/IFIP Network Operations and Management Symposium (NOMS 2022)*, pages 1–6, 2022.

- Frank Engelhardt, Sophie Herbrechtsmeyer, and Mesut Günes. Kinesthetic Coding Based on the Fast Wavelet Transform for Remote-Controlling a Quadrotor Drone. In *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, pages 157–162, 2022.

- Frank Engelhardt and Mesut Günes. Combined Certificate and Resource Discovery for Dynamically (Dis-)Aggregating IoT Processes. In Ralf H. Reussner, Anne Koziolek, and Robert Heinrich, editors, *INFORMATIK 2020*. Gesellschaft für Informatik, Bonn, January 2021.

- Frank Engelhardt, Johannes Behrens, and Mesut Günes. The OVGU Haptic Communication Testbed (OVGU-HC). In *31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications: Track 3: Practical and Experimental Systems (PIMRC'20 Track 3 Practical & Exper Sys)*, volume 31, London, United Kingdom (Great Britain), August 2020.

- Frank Engelhardt, Chenke Rong, and Mesut Günes. Towards Tactile Wireless Multi-Hop Networks - The Tactile Coordination Function as EDCA Supplement. In *2019 Wireless Telecommunications Symposium (WTS 2019)*, New York City, USA, April 2019.

- Frank Engelhardt, Johannes Behrens, and Mesut Günes. Demo - A Haptic Communication testbed - Integrating the Control Systems Domain Into Communication Testbeds. In *18. GI/ITG KuVS Fachgespräch SensorNetze, FGSN 2019*, pages 59–60. Otto-von-Guericke-Universität Magdeburg, 2019.

- Frank Engelhardt and Mesut Günes. Modeling Delay of Haptic Data in CSMA-based Wireless Multi-Hop Networks: A Probabilistic Approach. In *IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*, volume 30, pages 1–6. IEEE, September 2019.

- Frank Engelhardt and Mesut Günes. A Latency Analysis of IEEE 802.11-based Tactile Wireless Multi-Hop Networks. In *18 GI/ITG KuVS Fachgespräch SensorNetze, FGSN 2019*, pages 33–36. Otto-von-Guericke-Universität Magdeburg, September 2019.

- Frank Engelhardt and Mesut Günes. Haptic Communication Latency in Large-Scale Wireless Mesh Networks. In *17. GI/ITG KuVS Fachgespräch SensorNetze, FGSN 2018*, pages 15–18. Technische Universität Braunschweig, September 2018.

- Maik Riestock, Frank Engelhardt, Sebastian Zug, and Nico Hochgeschwender. User study on remotely controlled UAVs with focus on interfaces and data link quality. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3394–3400. IEEE, 2017.

- Maik Riestock, Frank Engelhardt, Sebastian Zug, and Nico Hochgeschwender. Exploring Gridmap-Based Interfaces for the Remote Control of UAVs under Bandwidth Limitations. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI 17, page 263264, New York, NY, USA, 2017. Association for Computing Machinery.

Glossary

5G 5th Generation Mobile Communication Technology

AC Access Category

AI Artificial Intelligence

AIFS Arbitration Inter-Frame Space

AN Access Network

AP Access Point

AR Augmented Reality

AS Autonomous System

BE Best Effort

BK Background

CCA Clear Channel Assessment

CDF cumulative distribution function

CDMA Frequency Division Multiple Access

CDN Content Delivery Network

CPS Cyber-Physical System

CSMA Carrier Sense Multiple Access

CSMA/CA CSMA with Collision Avoidance

DCF Distributed Coordination Function

DetNet Deterministic Networking

DoF Degree of Freedom

e2e end-to-end

EDCA Enhanced Distributed Channel Access

EDCAF Enhanced Distributed Channel Access Function

eMBB enhanced Mobile Broadband

ETSI European Telecommunications Standards Institute

HAVE Haptic Audio Visual Environment

HCF Hybrid Coordination Function

HDTF Haptic Digital Twin Framework

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IFS Inter-Frame Space

ISP Internet Service Provider

ITS Intelligent Transport System

ITS Intelligent Transportation System

ITU International Telecommunication Union

LiFi Visible Light Communication

LLC Logical Link Layer

LOS Line-of-Sight

MAC Medium Access Control

MCF Mesh Coordination Function

ME Matrix-Exponential

MIMO Multiple Input/Multiple Output

mMTC massive Machine-Type Communication

MPDU MAC Protocol Data Unit

MSDU MAC Service Data Unit

MSE Mean Square Error

NAV Network Allocation Vector

NCS Networked Control System

NGP Next Generation Protocols

NOMA Non-orthogonal Multiple Access

OVGU-HC Haptic Communication Testbed at the Otto-von-Guericke University of Magdeburg

OVGU-HCF OVGU-HC Framework

P-MSE Perceptual Mean Square Error

PaaS Platform-as-a-Service

PCF Point Coordination Function

PDF probability density function

PER Packet Error Rate

PMF probability mass function

QoE Quality of Experience

QoS Quality of Service

RMSE Rooted Mean Square Error

RTT Round-Trip-Time

SIFS Short Inter-Frame Space

SNIR Signal to Noise and Interference Ratio

TBMS Testbed Management System

TC Tactile Control

TCF Tactile Coordination Function

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

TI Tactile Internet

TSE Tactile Support Engine

TSM Tactile Service Manager

TSN Time-Sensitive Network

UP User Priority

URLL Ultra-Reliable and Low-Latency

URLLC Ultra-Reliable and Low-Latency Communication

VI Video

VLAN Virtual Local Area Network

VO Voice

VR Virtual Reality

WiFi Wireless Fidelity

WMHN wireless multi-hop network

WSN wireless sensor network

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe einer kommerziellen Promotionsberaterin/eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche der Urheberin/des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, 26. Oktober 2022

(Frank Engelhardt)