Queueing-Theoretic End-to-End Latency Modeling of Future Wireless Networks

Schulz

Technische Universität Dresden

Queueing-Theoretic End-to-End Latency Modeling of Future Wireless Networks

Philipp Schulz

der Fakultät Elektrotechnik und Informationstechnik der
Technischen Universität Dresden

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender:	Prof. Dr.-Ing. habil. Leon Urbas
Gutachter:	Prof. Dr.-Ing. Dr. h.c. Gerhard Fettweis
	Univ. Prof. Dr. techn. Markus Rupp
Tag der Einreichung:	01. Oktober 2019
Tag der Verteidigung:	13. Januar 2020

Philipp Schulz
Queueing-Theoretic End-to-End Latency Modeling of Future Wireless Networks
Dissertation, 13. Januar 2020

Vodafone Chair Mobile Communications Systems
Institut für Nachrichtentechnik
Fakultät Elektrotechnik und Informationstechnik
Technische Universität Dresden
01062 Dresden

Abstract

The fifth generation (5G) of mobile communication networks is envisioned to enable a variety of novel applications. These applications demand requirements from the network, which are diverse and challenging. Consequently, the mobile network has to be not only capable to meet the demands of one of these applications, but also be flexible enough that it can be tailored to different needs of various services. Among these new applications, there are use cases that require low latency as well as an ultra-high reliability, e. g., to ensure unobstructed production in factory automation or road safety for (autonomous) transportation. In these domains, the requirements are crucial, since violating them may lead to financial or even human damage. Hence, an ultra-low probability of failure is necessary.

Based on this, two major questions arise that are the motivation for this thesis. First, how can ultra-low failure probabilities be evaluated, since experiments or simulations would require a tremendous number of runs and, thus, turn out to be infeasible. Second, given a network that can be configured differently for different applications through the concept of network slicing, which performance can be expected by different parameters and what is their optimal choice, particularly in the presence of other applications.

In this thesis, both questions shall be answered by appropriate mathematical modeling of the radio interface and the radio access network. Thereby the aim is to find the distribution of the (end-to-end) latency, allowing to extract stochastic measures such as the mean, the variance, but also ultra-high percentiles at the distribution tail. The percentile analysis eventually leads to the desired evaluation of worst-case scenarios at ultra-low probabilities. Therefore, the mathematical tool of queuing theory is utilized to study video streaming performance and one or multiple (low-latency) applications. One of the key contributions is the development of a numeric algorithm to obtain the latency of general queuing systems for homogeneous as well as for prioritized heterogeneous traffic. This provides the foundation for analyzing and improving end-to-end latency for applications with known traffic distributions in arbitrary network topologies and consisting of one or multiple network slices.

Kurzfassung

Es wird erwartet, dass die fünfte Mobilfunkgeneration (5G) eine Reihe neuartiger Anwendungen ermöglichen wird. Allerdings stellen diese Anwendungen sowohl sehr unterschiedliche als auch überaus herausfordernde Anforderungen an das Netzwerk. Folglich muss das mobile Netz nicht nur die Voraussetzungen einer einzelnen Anwendungen erfüllen, sondern auch flexibel genug sein, um an die Vorgaben unterschiedlicher Dienste angepasst werden zu können. Ein Teil der neuen Anwendungen erfordert hochzuverlässige Kommunikation mit niedriger Latenz, um beispielsweise unterbrechungsfreie Produktion in der Fabrikautomatisierung oder Sicherheit im (autonomen) Straßenverkehr zu gewährleisten. In diesen Bereichen ist die Erfüllung der gestellten Anforderungen besonders kritisch, da eine Verletzung finanzielle oder sogar personelle Schäden nach sich ziehen könnte. Eine extrem niedrige Ausfallwahrscheinlichkeit ist daher von größter Wichtigkeit.

Daraus ergeben sich zwei wesentliche Fragestellungen, welche diese Arbeit motivieren. Erstens, wie können extrem niedrige Ausfallwahrscheinlichkeiten evaluiert werden. Ihr Nachweis durch Experimente oder Simulationen würde eine extrem große Anzahl an Durchläufen benötigen und sich daher als nicht realisierbar herausstellen. Zweitens, welche Performanz ist für ein gegebenes Netzwerk durch unterschiedliche Konfigurationen zu erwarten und wie kann die optimale Konfiguration gewählt werden. Diese Frage ist insbesondere dann interessant, wenn mehrere Anwendungen gleichzeitig bedient werden und durch sogenanntes Slicing für jeden Dienst unterschiedliche Konfigurationen möglich sind.

In dieser Arbeit werden beide Fragen durch geeignete mathematische Modellierung der Funkschnittstelle sowie des Funkzugangsnetzes (Radio Access Network) adressiert. Mithilfe der Warteschlangentheorie soll die stochastische Verteilung der (Ende-zu-Ende-) Latenz bestimmt werden. Dies liefert unterschiedliche stochastische Metriken, wie den Erwartungswert, die Varianz und insbesondere extrem hohe Perzentile am oberen Rand der Verteilung. Letztere geben schließlich Aufschluss über die gesuchten schlimmsten Fälle, die mit sehr geringer Wahrscheinlichkeit eintreten können. In der Arbeit werden

Videostreaming und ein oder mehrere niedriglatente Anwendungen untersucht. Zu den wichtigsten Beiträgen zählt dabei die Entwicklung einer numerischen Methode, um die Latenz in allgemeinen Warteschlangensystemen für homogenen sowie für priorisierten heterogenen Datenverkehr zu bestimmen. Dies legt die Grundlage für die Analyse und Verbesserung von Ende-zu-Ende-Latenz für Anwendungen mit bekannten Verkehrsverteilungen in beliebigen Netzwerktopologien mit ein oder mehreren Slices.

Acknowledgement

The research, which has lead to this thesis, has started around four years ago. During this time, I was accompanied by people without whom this thesis would not have been possible and whom I would like to thank.

First and foremost, my deepest gratitude is dedicated to my advisor Gerhard Fettweis. He provided the opportunity for starting and pursuing this adventure. With his valuable feedback and support, he helped me to stay on the right track as well as to develop not only technical but also organizational skills. Second, I would like to thank Markus Rupp for assuring his willingness and time for being a second reviewer of this thesis.

Furthermore, I appreciate the nice working atmosphere with former and current colleagues. Among them, I would like to highlight my gratitude for my group leaders Meryem and André for their guidance. In particular, I could always count on Meryem's feedback, even when she had a new position nine time zones away from me, and I appreciate her engagement for my two research stays in Berkeley. In a non-exhaustive list, I owe special thanks to Albrecht, Behnam, David, Henrik, Jay, Jens, Lucas, Maciej, Norman, Sandra, Sascha, and Tom. They enriched my time at the chair not only on a professional but also on a private level. I have learned a lot, especially through bringing the engineering perspective of my colleagues and my mathematical background together. In this regard, I would like to stress my gratitude for Henrik, who brought me into this topic, and for André, Lucas, Meryem, and Tom, who helped polishing the manuscript thanks to their sharp eyes and invaluable comments. Moreover, Eva, Kathrin, Raffael, Rüdiger, Steffen, Sylvia, and Yaning deserve many thanks for keeping the chair running and helping us in all administrative matters. As my work also depended on high performance computing, I also thank the ZIH staff for providing their resources and taking care of taurus.

My appreciation also goes to my partners in the project fast wireless. The collaboration was always fun and I received valuable feedback on my work. Exemplary, I would like to mention Andreas, Björn, Camilo, Dariush, Ines, Jens, Marco, Marcus, Markus, Sebastian, Tino, and Torsten.

I warmly thank my family and friends for always supporting and motivating me. Among them, I would like to highlight my special appreciation for René, who supported me with fruitful discussions. My particular gratitude is also dedicated to Susi, who was always there, when I needed somebody to talk. Last but definitely not least, I would like to thank my girlfriend Marietta, especially for being endlessly patient and believing in me.

Dresden, January 2020 Philipp Schulz

Contents

1 Introduction

The first chapter of this thesis is dedicated for motivating the mathematical modeling conducted in this work by envisioned applications, which may benefit from accurate models. Furthermore, the state of the art is presented and the contributions of this thesis are provided as an outline.

1.1 Motivation

When the work on this thesis started, the fifth generation (5G) of mobile communication networks was still a pure vision. Its development was mainly driven by the use cases and their requirements, which are targeted for 5G networks. Today, the first operators claim that they start rolling out 5G networks. They still have to prove to which extend the ambitious demands can already be met. In the end of 2008, when the current standard Long-Term Evolution (LTE) was introduced [3GP08a], it turned out that the objectives for fourth generation (4G) mobile communications systems, which were defined by the International Telecom Union (ITU) in [ITU08], could not be achieved completely. Thus, LTE is often referred to as 3.9G. It still took until the introduction of LTE-Advanced [3GP11b] in 2011 to have an actual 4G standard. Possibly, a similar story may happen with the early standards and roll-outs of 5G, such that there remains much work towards the realization of 5G networks at its full extend. This thesis is motivated by 5G applications and its requirements. Thereby, the focus is led by understanding the technical necessities of applications which require low latency and, thus, on the mathematical modeling and optimization of latency in mobile networks. The envisioned 5G applications are presented in the following.

1.1.1 Use Cases and their Requirements

As visualized in Fig. 1.1, the envisioned 5G use cases can be devided into three categories [ITU15], namely enhanced mobile broadband (eMBB), massive

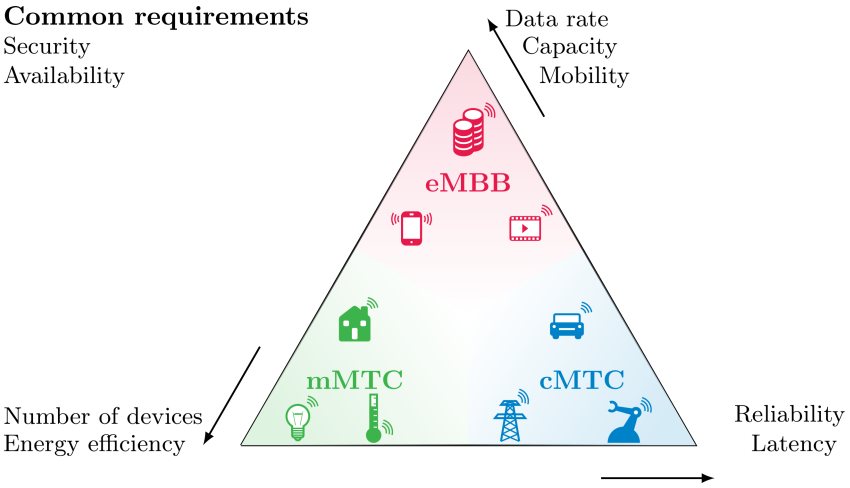


Fig. 1.1. 5G use cases with their key requirements according to [ITU15; MET16].

machine-type communication (mMTC), and critical machine-type communication (cMTC)[1]. They are primarily characterized by their requirements and have smooth transitions in between. The applications of eMBB principally comprise multimedia and entertainment applications, which mainly require enhancing the key performance indicators (KPIs) throughput and capacity, which were the main drivers for previous generations as well. Herein, humans are connected to each other or to multimedia services. With the introduction of 5G it is expected that machines use mobile connections to communicate to each other as well, leading to mMTC and cMTC. The domain of mMTC is characterized by applications that connect a massive number of sensors and actuators. Here, throughput is less important, because often only a small amount of data has to be transferred. Instead, the network has to support a huge number of devices, provide a wide coverage, and should allow energy-efficient operation, since battery life is an important issue. In contrast, cMTC refers to mission-critical applications with demanding requirements (mostly) on the KPIs latency and reliability. Thus, the term ultra-reliable low latency communication (URLLC) has been established to refer to those applications and the communication technologies which enable them.

This thesis focuses on cMTC or URLLC and in particular on the latency aspects with respect to queuing effects. In our work [Sch+17], a requirement analysis

[1]Different terms have been established. For instance, the ITU denotes the cMTC applications as URLLC in [ITU15]. In contrast, the well-known project METIS refers to those use cases as ultra-reliable machine-type communication (uMTC) in their deliverable [MET16]. In this thesis, cMTC is used as the umbrella term (complementing mMTC), and URLLC refers to applications which require both, low latency and high reliability.

was conducted, comprising a non-exhaustive list of latency-critical applications. Its results are summarized in Table 1.1.

Among those use cases, applications from factory automation are identified as the most demanding in terms of latency and reliability. Factory automation denotes discrete manufacturing, which comprises producing, testing, and packaging of products in multiple, discrete steps. Those steps typically require sensors and actuators to be connected to controllers in real-time. Replacing conventional wired links with wireless solutions is expected to bring several advantages, such as easy installation, reduced attrition, and increased mobility. However, a latency down to $0.25\,\mathrm{ms}$ and a packet loss rate (PLR) of 10^{-9} as a reliability metric are required. The table also provides more detailed examples with their respective requirements.

Other important applications can be found in the field of intelligent transport systems (ITSs). Here, the aim is to increase safety and efficiency for road traffic through communications between vehicles Compared with factory automation, the demands on latency and reliability are more moderate with a required delay down to $10\,\mathrm{ms}$ and an accepted PLR down to 10^{-5}. However, increased mobility and communication distances pose additional challenges on the communication system.

Table 1.1 is complemented by the further use cases process automation, smart grids, and professional audio along with their respective requirements.

1.1.2 Low Latency and High Reliability

URLLC applications usually require both low latency and ultra-high reliability. Often both metrics are mutually connected. For instance, a packet that exceeds a required latency bound may become useless for a latency-critical application and, thus, can be considered as lost. Thereby, harming the latency threshold would increase the PLR. On the other hand, a high latency could be tolerated, if it occurs only very rarely, such that the PLR is not violated (together with other reasons for packet loss). Therefore, it is reasonable to consider the probability distribution of the experienced latency and require that the latency does not exceed a certain threshold even for very high percentiles. Indeed, the ITU requires 5G to deliver a $32\,\mathrm{Byte}$ packet within a latency of $1\,\mathrm{ms}$ and a residual error of 10^{-5} [ITU17]. Even more strict requirements were formulated in Section 1.1.1.

Tab. 1.1. Requirement analysis for anticipated URLLC use cases from our work [Sch+17].

Use case	Latency (ms)	Reliability (PLR)	Update Time (ms)	Data Size (Byte)	Device Density	Communication Range (m)	Mobility ($km\,h^{-1}$)
Factory automation[a]	0.25 to 10	10^{-9}	0.5 to 50	10 to 300	0.33 to $3\,m^{-2}$	50 to 100	<30
Manufacturing cell	5	10^{-9}	50	< 16	0.33 to $3\,m^{-2}$	50 to 100	<30
Machine tools	0.25	10^{-9}	0.5	50	0.33 to $3\,m^{-2}$	50 to 100	<30
Printing machines	1	10^{-9}	2	30	0.33 to $3\,m^{-2}$	50 to 100	<30
Packaging machines	2.5	10^{-9}	5	15	0.33 to $3\,m^{-2}$	50 to 100	<30
Process automation[b]	50 to 100	10^{-3} to 10^{-4}	100 to 5000	40 to 100	1×10^4 per plant	100 to 500	<5
Smart grids[c]	3 to 20	10^{-6}	10 to 100	80 to 1000	10 to $2000\,km^{-2}$	a few m to km	0
ITS[d]	10 to 100	10^{-3} to 10^{-5}	100 to 1000	500 to several MB	500 to $3000\,km^{-2}$	200 to 2000	50 to 500
Road safety urban	10 to 100	10^{-3} to 10^{-5}	100	< 500	$3000\,km^{-2}$	500	<100
Road safety highway	10 to 100	10^{-3} to 10^{-5}	100	< 500	$500\,km^{-2}$	2000	<500
Urban intersection	100	10^{-5}	1000	1 MB per car	$3000\,km^{-2}$	200	<50
Traffic efficiency	100	10^{-3}	1000	1 kB	$3000\,km^{-2}$	2000	<500
Professional audio[e]	2	10^{-6}	0.01 to 0.5	3 to 1000	up to $1\,m^{-2}$	100	<5

[a] Primary source for factory automation: [Fro+14].
[b] Primary source for process automation: [Sve11].
[c] Primary source for smart grids: [IEC13].
[d] Primary sources for ITS: [5GP15; ETS09; Ham14].
[e] The data for professional audio stems from a requirement analysis of the project partner Freedelity.

When it comes to critical communications, violations of the stated requirements may lead to failure of the application and, thus, to severe financial or even human damage. Hence, it is of utmost importance that meeting the requirements can be guaranteed to a certain extent, i. e., up to a very high probability as agreed when starting the URLLC service. Simulations or even real experiments can be used to study system behavior but are not appropriate to obtain such guarantees, because a tremendous number of runs would be required to have statistical significance for the tail of the distribution as well. Here, accurate mathematical models, as proposed in this thesis, can be an efficient and fast alternative.

Recent work, e. g., [SGA15; SYQ17], focused on latency modeling with respect to URLLC. The authors in [SGA15] investigate wireless fading channels with finite block length channel coding. In [SYQ17], the need for latency modeling for URLLC traffic is stated and queuing delay violation is identified as an important issue for reliability. Here, queuing delay refers to the time, when data has to wait for competing data due to a limited amount of resources. In addition, the comprehensive survey conducted in [Bri+16] provides a broad overview of causes for delay and approaches to reduce latency in the Internet. According to the survey, queuing delay accounts for the largest portion to the transmission latency and most of the latency is contributed at the network edge. For these reasons, this thesis focuses on modeling latency due to queuing effects.

Some methods to achieve low latency in 5G are summarized in our work [Sch+19b]. For instance, subdividing the transmission time interval (TTI), i. e., using less symbols per TTI, leads to a shorter waiting time for the next TTI, a shorter transmission time itself and shorter round trip times for possible retransmissions. In addition, the novel radio access technology (RAT) new radio (NR) introduces higher options for sub carrier spacing (SCS), which enable shorter symbol lengths and hence shorter TTIs. The new standard also requires user equipment (UE) to implement shorter processing times for encoding and decoding. Furthermore, uplink (UL) resources can be preallocated for URLLC devices, such that they save the time for the otherwise necessary scheduling request and waiting for the grant.

To ensure high reliability, also other reasons, different from losing packets due to high delay, have to be taken into account. Packets can get lost or become erroneous while being transmitted. For wireless systems the reasons for that include path loss, fading, shadowing, and interference. A promising way to cope with this is to introduce diversity, which means that information is being transmitted redundantly in the time, frequency, or spatial domain. In the context of

URLLC, where latency is critical, time diversity, i. e., performing retransmissions, usually is not feasible. Instead, multi connectivity (MC), i. e., having multiple connections in parallel, is preferred. Those links can be established on different frequencies or between multiple antennas located at one or different sites (i. e., spatial diversity).

Other reliability aspects that are not related to failure through outdated packets are out of scope of this thesis. Instead, the reader is referred to our joint work in this area, which is shortly summarized in the remainder of this section. In [Höß+19b], reliability theory is applied to wireless communications. Terms like reliability and availability are often loosely utilized in the wireless community. In our work, PLR is identified as not being a sufficient metric to characterize reliability as it does not reflect the time dimension and, thus, rather refers to availability.[2] Instead the more meaningful term *mission availability* is introduced, which also considers the duration of an application and the fact that applications can often tolerate short packet loss. A theoretical framework to analyze MC scenarios for different combining schemes is presented in [Wol+17; Wol+18; Wol+19]. For instance, the framework allows to study the trade-off between diversity and multiplexing, i. e., between reliability and throughput. In [Höß+20], a many-to-many matching algorithm was used to cope with the problem of allocating multiple channels, i. e., resources, to multiple users. Furthermore, the work in [Höß+19a] introduces a dynamic connectivity scheme, which leads to efficient usage of resources for applications that can tolerate short outage through switching the channel in case of bad conditions. Finally, a scheme that adapts the number of links in order to reduce consecutive packet loss is proposed and analyzed in [Sch+20a].

1.1.3 Network Slicing

One major challenge of 5G is that it aims not only at targeting one specific use case, but should address all of the dimensions mentioned in Section 1.1.1 with their diverging requirements. As each domain is challenging for itself, satisfying all requirements at once appears utopian. Instead, a flexible network that can be tailored to the different use cases constitutes a promising solution.

In their white paper [NGM15], the Next Generation Mobile Networks (NGMN) alliance proposes the concept of network slicing for 5G. The white paper is

[2] Availability can be defined as the complement of the PLR, i. e., the probability of successfully transmitting a packet corresponding to $1 - \text{PLR}$.

complemented by their deliverable [NGM16]. A network slice essentially denotes a logical network, which is established to serve a use case. To meet the specific requirements, the network slice handles the control and user plane for one service in a particular way, by using a collection of 5G network functions and specific RAT settings. This can span all domains of the network, i. e., different software modules running on cloud nodes, specific configurations of the transport network supporting flexible locations of network functions, dedicated radio parameterization or even a specific RAT, and the configuration of the UE itself. Network slicing is enabled by the concepts network functions virtualization (NFV) and software-defined network (SDN) [Sim+17], which allow for flexible placement of network functions as well as flexible and scalable configuration of the network, respectively. Furthermore, mobile edge computing (MEC) provides flexible placement of computing resources close to the network edge.

However, the question still remains, how such network slices can be parameterized optimally and which performance can be expected then, especially with dynamically changing conditions. In this regard, the network slicing approach can benefit from appropriate performance modeling, which can provide the answers. Consequently, also models supporting multiple slices are introduced in this thesis.

1.2 State of the Art

The current generation of mobile communications systems 4G, i. e., LTE (Advanced), is by far not capable to meet the desired requirements in Table 1.1. In our work [Sch+17], measurements that were performed in a live network in 2015 revealed that only an average round trip time (RTT)[3] of $47\,\mathrm{ms}$, $43\,\mathrm{ms}$, and $35\,\mathrm{ms}$ can be achieved, when connecting to a server in the Internet, to the core network gateway, or to a dynamic host configuration protocol (DHCP) server in the core network, respectively, indicating room for latency improvement in the radio access network (RAN). Our study also showed that the latency is very sensitive to the network load, since the latency significantly degrades during rush hours, when the average latency to an internet server increases up to $80\,\mathrm{ms}$.

As for the latency modeling, different approaches are being pursued. For instance, latency can be measured empirically in live networks and curve fitting can be applied to obtain stochastic distributions of the latency (e. g., [VBK18]). This

[3] The RTT denotes the time it takes to send data from a sender to a receiver and back. In this study it was measured by performing Internet control message protocol (ICMP) pings.

approach may lead to good approximations of the reality, but neither provides functional relationships to the network parameters or any reasoning for the resulting distributions, nor it is applicable on other networks as it depends on the measurements. In addition, this approach is not appropriate for good approximations of the distribution tail due to a limited number of measurements. Furthermore, there exist relatively simple analyses based on the standardized numerology to estimate RAN delay. As an example, the author of [Abe10] claims that it was already verified by the feasibility study [3GP08b] of 3rd Generation Partnership Project (3GPP) that the radio interface of LTE (Release 8) achieves a latency target of $5\,\mathrm{ms}$. However, those evaluations only take the foreseen numerology into account, consider only one possible retransmission up to a assumed probability, and are based on the assumption of an unloaded network, to derive average latency values. This approach can neither provide any worst case results nor it evaluates any effects from random user behavior and network load.

To capture the random effects as well, e. g., in radio conditions or in data traffic, stochastic models are necessary. Therefore, the most common approaches are *queuing theory* (e. g., [Asm03; Kle75; Kle76]) and *network calculus* (e. g., [LT04; LJ08]). Both theories aim for analyzing systems with shared resources and the impact of the resource sharing on KPIs, such as latency, capacity, or dropping rates. Whereas queuing theory focuses on determining stochastic measures (e. g., moments) or even the distribution of the KPIs, depending on analytical tractability, the approach of network calculus rather provides upper performance bounds by considering cumulative functions for arrival and service as well as a min-plus algebra. The bounding functions are traditionally deterministic [LT04], but the framework has been extended to stochastic network calculus [LJ08]. In other words, queuing theory may refer to stochastic analysis of the system, whereas network calculus conducts a worst case analysis.

Both approaches, queuing theory and network calculus, have been widely used to analyze (wireless) communications systems. For instance, network calculus was used in [Fid06; ALB16; SGA15] to derive probabilistic performance bounds for the latency of wireless fading channels. In particular, [Fid06] was one of the first works that applied network calculus to wireless fading channels, [ALB16] studied multihop fading channels with frequency hopping, and in [SGA15], channel coding at finite block length is considered. The authors of [Lei+11] analyzed an LTE access network with real-time and non-real-time traffic with the help of stochastic network calculus for exponentially bounded arrival and

service processes. However, the results are very coarse with end-to-end (E2E) latency values of $1.5\,\mathrm{s}$ that are exceeded with a violation probability of $20\,\%$ for real-time traffic, which is far from the requirements and actually achievable performance.

The application of queuing theory [BBP04; Bon04b; Bon05; Kle+16; KGF14; KFF14; KF15] and of queuing networks [Küh76; Küh15; Jar+11; Mah+14; Mah+15; ZNS14] in particular will be discussed in detail in the Sections 2.2.2 and 2.3.2, respectively. In summary, it can be said that the existing work mainly aims for obtaining average values (in some cases including the coefficient of variations) or performance bounds. In addition, not all of the related work covers latency, but rather focuses on other metrics, such as throughput, blocking probability, or capacity. Furthermore, the cited works are often restricted to simple arrival and service models and consider homogeneous traffic.

1.3 Contribution

The overall objective of this thesis is to develop foundations for a flexible mathematical framework to conduct E2E latency analysis of one or multiple 5G use cases. More specifically, the state of the art is being complemented by the following contributions with this thesis:

- In the context of this thesis, a MATLAB [Mat19] class was implemented that simplifies the handling of random variables (RVs) and their distributions. In contrast to the already existing distribution class in MATLAB, whose purpose is mainly the distribution fitting, sampling, and statistical analysis, the new class is more tailored to performing operations on (independent) RVs. The main idea is to apply analytical, i. e., explicit, formulas wherever possible and to fall back to numerical methods to be generally applicable (cf. Section 2.4).
- The wireless downlink (DL) access is modeled with the help of flow-level queuing models (cf. Section 3.1), which are based on existing work. This model is the foundation for extending existing models for the performance evaluation of video streaming traffic. The extension is constituted by incorporating multi-cellular interference dynamics and by applying a finite volume method (FVM) to obtain a numerical solution for the arising partial differential equation (PDE) systems. Thereby, the streaming KPIs startup delay distribution and buffer starvation probability are derived, referring

to latency and reliability in the context of video streaming (cf. Section 3.2, [SKF17; Sch+20b]).

- An approach to transfer the existing DL flow-level models to study the UL as well is proposed (cf. Section 3.3).
- Queuing networks are proposed to conduct E2E latency analysis in mobile communication networks, providing a flexible framework for arbitrary network topologies. Such a tool offers valuable insights on the impact of different network parameters and thereby enables finding adequate configurations for networks or network slices. Thereby, foundations for (autonomous) network optimization are laid (cf. Section 4.2, [Sch+19c]).
- As an important ingredient for a general performance evaluation framework (cf. previous item), general queuing systems, i. e., $\mathrm{GI}/\mathrm{GI}/1$ queues, have to be analyzed. Therefore, a numerical method is proposed to conduct this analysis for general independent arrival and service processes. The results are complemented by an existing expression that bounds $\mathrm{GI}/\mathrm{GI}/1$ latency (cf. Section 4.3, [Sch+19a]).
- To provide also analysis tools for heterogeneous traffic of different applications that is handled by different slices, different schedulers are integrated into the model to study prioritized heterogeneous traffic as well (cf. Section 4.4, [Sch+19a]).
- The proposed models were compared to the respective simulations in terms of computational effort and complexity. It turned out that the developed models can be significantly faster than simulations (cf. Sections 3.4, 4.6).
- Furthermore, performance with respect to latency and throughput is improved for a cellular scenario with prioritized traffic of two classes. (cf. Chapter 5).
- Finally, open questions are highlighted at the end of Chapters 3–5, providing starting points for further research.

2

Modeling Approaches for Communication Networks

In general, for the evaluation of a wireless network's performance, there are three kinds of approaches. First, either an existing working environment (e. g., a real cellular network) or a *testbed* can be set up for conducting measurements of the KPIs of interest. By its nature, this leads to the most realistic results, but it is not very flexible and can be expensive, time-consuming or even infeasible. Especially when the technology is not fully matured yet, appropriate hardware is not implemented yet, or when data is confidential, experiments are not a possible choice. Also, new algorithms should usually not be tested in a live network, since they may fail.

Secondly, the performance can be estimated by running *simulations* and trying to map the reality onto software. Thereby, simulations always represent an abstraction of reality and differ in the level of details that are incorporated. Thus, one distinguishes for instance between link-level or system-level simulations. Whereas the former approach simulates the wireless channel in full detail and, thus, offers a microscopic view, the latter has an abstracted view on the transmission itself (e. g., with knowledge gained from the link level) and focuses more on the entire network for a macroscopic perspective [Cab12]. Compared to experiments, simulations can be parameterized and parallelized very easily. However, even with the availability of today's high performance computing, they can become infeasible when URLLC requirements (e. g., PLR of 10^{-9}) demand a tremendous number of simulation runs to observe ultra-low outage with statistical relevance.[1]

The third approach, *mathematical modeling*, pushes the level of abstraction even further. Here, stochastic models are involved as a powerful tool to capture the random behavior of users and physical conditions. The aim is to find functional

[1] As the probability is very low, only a few values are available at the tail of a distribution, making estimations of the distribution in this region inaccurate. For an example, the reader is referred to Fig. 4.10. There it can be observed how the results for extremely high percentiles may vary.

relationships between the scenario parameters and KPIs that approximate the real performance sufficiently accurately to obtain valuable insights. Even though those relationships may be too complex to be solved analytically, they can usually be treated by numerical methods, while still being faster than simulations. However, models will always be only an abstraction of the reality, because not everything can be considered and, thus, they may be oversimplified.

This thesis mainly focuses on the third approach by providing such mathematical models for the E2E latency in wireless networks and by evaluating them. Such evaluations rely on appropriate assumptions and knowledge about the (random) traffic behavior of the considered applications and its users. Therefore, this chapter provides foundations for spatial and temporal traffic modeling and introduces the mathematical framework of queuing systems and queuing networks. Lastly, a class implemented in MATLAB [Mat19] to handle mathematical distributions is presented, which turned out to be of key importance in this work.

2.1 Traffic Modeling

No matter whether it comes to simulations or mathematical modeling, realistic assumptions on user behavior (macroscopic) and/or data traffic characteristics (microscopic) have to be stated as an input for the simulations or as a starting point for the modeling, since the underlying traffic demand influences the system dynamics. The mentioned levels of detail refer to two different time scales and hence to traffic modeling on *flow-level* or *packet-level*, respectively. Here, the notion of a data flow [Rob01] comprises all packets belonging to the same object, such as a web page, a video, or any other file. They can be described by stochastic processes, characterizing the traffic behavior in the spatial as well as in the temporal domain. This comprises the random process of inter-arrival times T (cf. Section 2.2.1) between users or packets, and the respective flow or packet sizes.

Usually, models establish a trade-off between accuracy and simplicity (and thereby analytical tractability). However, the proposed numerical method in Section 4.3 can also deal with general models.

2.1.1 Temporal Traffic Modeling

The temporal behavior of data traffic strongly depends on the considered application. Modeling usually relies on knowledge or assumptions about this application

and is often supported or validated by empirical data. The NGMN alliance provides a comprehensive list of detailed temporal traffic models for typical 4G use cases, i. e., file transfer protocol (FTP), web browsing, video streaming, voice over Internet protocol (VoIP), and gaming, in [NGM08a]. However, for the three envisioned fields of 5G use cases (cf. Section 1.1.1), those may need revision and amendments. Since mobile broadband (MBB) applications have been introduced already with third generation (3G), more studies and data are available than for the new fields of mMTC and cMTC. Laner et al. (e. g., [Lan+12; Lan+13b; Nik+13; Lan+13a]) conducted extensive studies in this research area.

Mobile Broadband Traffic

Comprising mostly multimedia applications, broadband traffic is usually characterized by large file sizes and session durations. Therefore, it is reasonable to describe the traffic on flow-level. In this regard, it is a common assumption to consider (inter-) arrivals to stem from a memoryless[2] random process, i. e., inter-arrival times follow an exponential distribution. The authors in [Lan+12] confirmed this assumption with data from a real 3G network in Viena. The same study also reveals an exponential distribution of session durations on logarithmic scale, resulting in a heavy tail distribution. The authors identify the mixture of different traffic types as a reason.

If a particular application is considered, more specific models can be found. For instance, in video streaming, the (view) duration is an important characterization for the data traffic. The authors in [Xu+13] assume exponentially distributed video durations and claim that this already reveals essential features of the system. Thanks to the memorylessness of the exponential distribution, the system becomes analytically tractable in a simple way. However, in their later work [Xu+17], they propose a hyper-exponential distribution with two phases instead, which is based on an empirical study. The two phases represent a mixture of two classes of viewing durations, namely short and long viewing time.

Descriptions also exist on packet-level. [NGM08a] models video streaming traffic with periodic arrivals of frames, consisting of eight packets, whose sizes follow a truncated Pareto distribution. The inter-arrival times of the frame's slices are

[2] A RV X referring to a time is called *memoryless*, if and only if (iff) its probability distribution does not depend on how much time has elapsed already, i. e., $\mathbb{P}[X > t + \Delta t \mid X > t] = \mathbb{P}[X > \Delta t]$. The only distributions that satisfy this condition are the exponential distribution $\mathrm{Exp}(\lambda)$ for continuous RVs [PP02, Eq. (4-32)] and the geometric distribution $\mathrm{Geo}(p)$ for discrete RVs [PP02, Example 4-16].

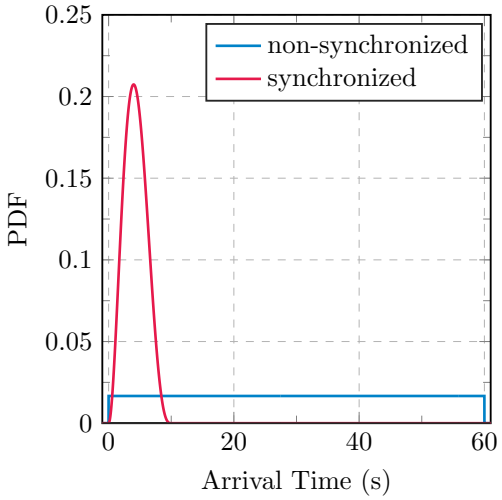


Fig. 2.1. Illustration of 3GPP traffic models for mMTC [3GP11a].

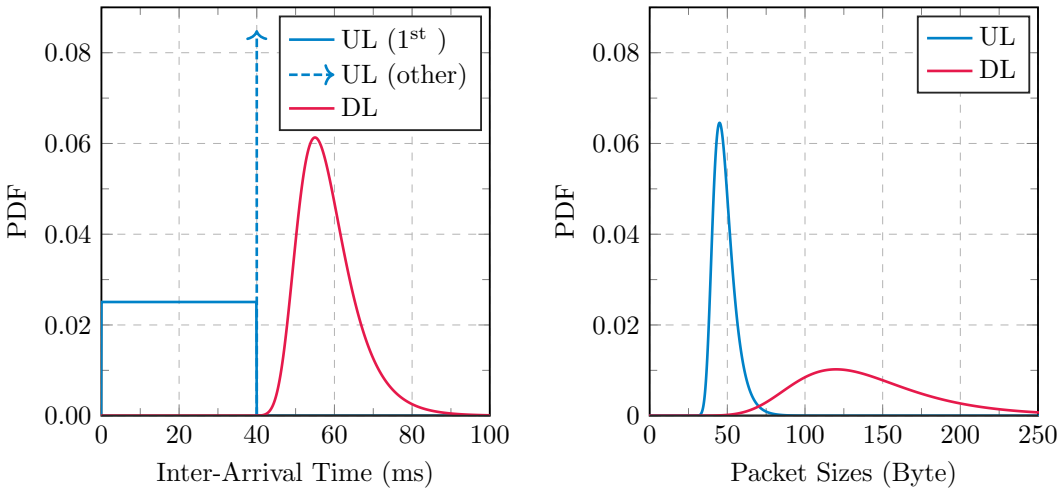
assumed to originate from another truncated Pareto distribution with different parameters.

Massive Sensor Traffic

In the technical report [3GP11a], 3GPP specifies two traffic models for machine-type communication (MTC). Here, they refer to mMTC and do not consider any critical communications. Two different situations are taken into account. First, under the assumption of non-synchronized devices, the arrivals are assumed to be uniformly distributed over an interval $[0, t_{\text{async}}]$, which is denominated here as $\mathcal{U}(0, t_{\text{async}})$. Second, if devices are synchronized, the arrivals are modeled by a beta distribution $\mathrm{B}(3, 4)$ scaled onto the interval $[0, t_{\text{sync}}]$, which constitutes a peak[3] at $\frac{2}{5}t_{\text{sync}}$, around which the synchronized arrivals are spread. For the synchronous case a much shorter interval bound is considered ($t_{\text{sync}} \ll t_{\text{async}}$). Both models are illustrated in Fig. 2.1.

Unfortunately, the 3GPP model is confronted with two issues. First, the report does not provide any reasoning for the choice of the beta distribution and its parameters. Second, it only provides the distribution of the arrival times rather than inter-arrival times, making it analytically hard to handle. The first gap is closed by [CSP16]. The authors justify the beta distribution with an alarm propagation model. However, they also notice that the parameters α, β, and t_{sync}

[3]The *mode*, i. e., the value that maximizes the probability density function (PDF), of a beta distribution [PP02, Section 4.3] $\mathrm{B}(\alpha, \beta)$ with shape parameters $\alpha, \beta > 1$ is located at $\frac{\alpha-1}{\alpha+\beta-2}$. This can be simply obtained by finding the root of the first derivative of the PDF and showing that the second derivative is negative in this point.



(a) Inter-arrival time distribution

(b) Packet size distribution

Fig. 2.2. Illustration of the NGMN traffic model for gaming [NGM08a].

have to be adjusted to the scenario at hand. The second problem is addressed in [Lan+13b] by proposing a modulated Poisson process to obtain an appropriate approximation of the inter-arrival time distribution.

According to [3GP11a], the packet size is fixed to 200 bytes. However, the Internet of things (IoT) may create more diverse packet sizes, due to the variety of imaginable sensors. Together with more sophisticated arrival models, this creates further research potential.

Mission-Critical Traffic

In former generations of mobile communications systems, cMTC were not foreseen. However, [NGM08a] already defined the interactive real-time use case gaming. Historically, since there are multiplayer games over networks with short reaction times, gamers are among the first users who suffered from large latency. Thus, it is not a surprise that the characteristics of gaming traffic appear appropriate also for cMTC. According to [NGM08a], UL traffic behaves differently from DL. After an initial uniformly distributed first arrival time, the UL arrivals are assumed to be periodic, whereas DL packet inter-arrival times always follow a Fisher-Tippet distribution[4]. The packet size here is also assumed to be Fisher-Tippet distributed (rounded with the floor function) with different parameters for both directions. The model is illustrated in Fig. 2.2 by showing the considered PDFs.

[4]The Fisher-Tippet distribution is also known as the generalized extreme value distribution [Col01, Section 3.1.3].

Tab. 2.1. Overview about traffic models in standardization on packet-level.

	eMBB	**mMTC**		**cMTC**	
		Option 1	Option 2	Option 1	Option 2
Purpose	video	asynchronous sensors	synchronized sensors	n/a	n/a
Arrivals	periodic / truncated Pareto	$\mathcal{U}(0, t_{\text{async}})$	$\mathrm{B}(3, 4)$ on $[0, t_{\text{sync}}]$	periodic	Poisson
Size	truncated Pareto	fixed, 200 bytes		fixed, $\{32, 50, 200\}$ bytes	
Reference	[NGM08a]	[3GP11a]		[3GP17]	

However, the described gaming model is based on former cellular technology and therefore based on long cycles of around $50\,\mathrm{ms}$. For new applications, 3GPP defined simulation assumptions for URLLC in their technical report [3GP17]. There, only simple models with either periodic (Option 1) or Poisson (Option 2) arrivals and different fixed packet sizes are considered. As with cMTC, realistic modeling of URLLC traffic still constitutes great potential for further research.

Temporal traffic models on packet-level from NGMN and 3GPP for all three 5G application domains are summarized in Table 2.1.

2.1.2 Spatial Traffic Modeling

For the evaluation of cellular network performance, it is not only important to know when data is demanded, but also where. In particular, local hotspots may lead to local cell congestion and thus, to performance degradation. However, mainly due to privacy issues, it is cumbersome to obtain empirical data to develop spatial traffic models, as shown in the following.

Empirical Data

For this purpose, two data sets were gathered and analyzed[5]. The first data set, was obtained from crowd-sourced data of a traffic monitoring app [Kun16]. Anonymous data of user sessions were collected containing global positioning system (GPS)-based location data, start time and duration, as well as the data volume in UL and DL. The database comprises around 500,000 entries for the

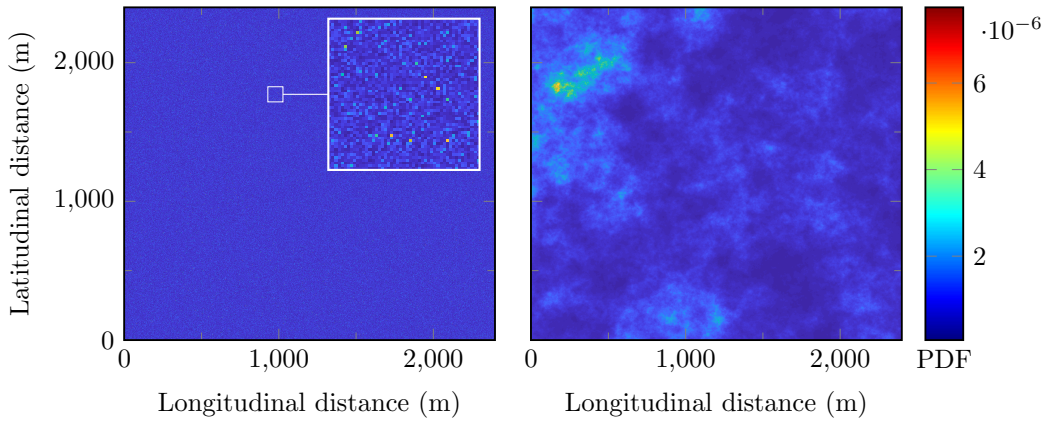
[5]The presented work are results from a joint supervision by Dr. Henrik Klessig and the author of this thesis of the diploma theses of Lucas Scheuvens (née Schwartz) [Sch16] and Henning Kuntzschmann [Kun16].

city of Berlin and 50,000 entries for the city of Madrid, which were both collected within three months in 2015.

The second data set pursued a different approach by collecting social media data that contains geographic information [Sch16]. Since actual mobile data is not always accessible, the idea was to study whether publicly available social media data is appropriate to estimate traffic demand. If this hypothesis turns out to be valid, social media data could serve as a complementary input for self-organizing network (SON) algorithms (cf. Chapter 5). The data were fetched from the social network Twitter over four weeks. In contrast to the first data set, the entries do not contain any information about duration and data volume. Still it may provide insights about user locations and the intensity of their device usage. For the data analysis, the Manhattan region was chosen, due to the high density of tweets, with approximately 376,000 tweets in total. Furthermore, the cities of Berlin and Madrid with about 45,000 and 109,000 tweets, respectively, were analyzed for comparison with the other data set.

Indeed, in our joint work [Kle+17], a strong linear temporal correlation between the tweet number and the overall traffic volume could be identified, which suggests social media data being a good indicator for the traffic load. Furthermore, a moderate to high positive spatial correlation was observed between both data sets, especially for coarser resolutions (i. e., $\sqrt{\Delta x \Delta y} \geq 50\,\mathrm{m}$ with $\Delta x, \Delta y$ being the grid constants in longitude and latitude, respectively), which is assumed to be a result of the limited sample size in both data sets. Here, the correlation according to Spearman [Jan11] turns out to be much higher than Pearson's correlation coefficient [Jan11], indicating a monotone, non-linear relationship between both data sets.

Motivated by the observed correlation, an approximate mapping between the Twitter data and the mobile data was defined in [Kle+17], so that the twitter data can serve as an input for the queuing-theoretic performance evaluation framework introduced in [Kle+16]. As a result, it turned out that the available Twitter data provides reasonable results for sufficiently large inter-site distances, due to the sparsity of the data. However, since Twitter is not the only social network and since in future there may be even more available data sources (e. g., from IoT devices), chances are high that aggregating such sources will result in suitable traffic demand estimators.



(a) No spatial correlation (b) With spatial correlation

Fig. 2.3. Illustration of the spatial traffic model. (a) If the spatial correlation was not taken into account, only noise would be sampled, which is only distinguishable in the enlarged extract. (b) Example realization of the traffic model in [Sch16].

Modeling

In [Sch16], a statistical analysis and modeling of the spatial data was conducted as well. It was observed that the spatial distribution of the tweet density can be approximated well by a log-normal distribution, especially when the considered region is restricted to a homogeneous region (e. g., with respect to demographic data). This is in particular interesting, because the spatial distribution of mobile data traffic can also be approximated by a log-normal distribution (e. g., [Lee+14]).

However, in [Sch16] the expected spatial autocorrelation of the data set was confirmed and quantified. Thus, it is not sufficient to sample independently from a log-normal distribution for the generation of a realistic traffic map realization. To create a realization that exhibits the same statistical properties, the author of [Sch16] applied a method called *circulant embedding of the covariance matrix* [DN97] with reasonable computational complexity. Therefore, the data was transformed from the log-normal domain to the normal domain and analyzed for its spatial correlation. With this information and the help of circulant embedding, a realization can be created in the normal domain and transformed back into the log-normal domain. An example realization is depicted in Fig. 2.3b. For comparison, a realization without any spatial correlation was created as well (Fig. 2.3a), which just appears as noise and does not constitute any hot spots.

2.2 Queuing Systems

Whenever resources are shared in a system, performance may be degraded due to the limited number of resources. A real-world example may be one or multiple counters (i. e., *servers*), where clients (i. e., *mobile devices*) form queues to be served one after the other. Since clients will arrive at the counter randomly and each may have a different number of goods in his or her basket, such that also the service takes a random time, this scenario is governed by two random processes.

As a mathematical tool, queuing theory was introduced (e. g., [Kle75]) to analyze such systems stochastically and to derive performance measures, in particular with respect to *waiting* or *sojourn time*, *throughput* and *capacity*. Here, the sojourn time denotes the entire time a client spends in the system, comprising the time for his or her own service and the time waiting for the service of other clients, which is referred to as waiting time. Because it provides a powerful tool, queuing theory gained a lot of attention also in (wireless) communications (cf. Section 2.2.2).

It should be noted that queuing has to be interpreted from a very general perspective. The "clients" can be anything, such as users, files or single packets. For general descriptions this work just refers to them as objects. Also the "servers" do not necessarily refer to physical machines but can also be the cores of a processor or the available resource blocks (RBs) at a base station (BS). Furthermore, the example of a supermarket counter should not mislead to the understanding that the queued elements are always served one after the other. There are also models were multiple objects can be served in parallel (even with only one server being available).

2.2.1 Definitions and Basic Concept

Usually, a *queue*, as illustrated in Fig. 2.4, is characterized by the following properties:

Inter-arrival time distribution. The inter-arrival time T_n is an RV that describes the time difference between two consecutive arrivals (i. e., between the $(n-1)^{\text{th}}$ and n^{th} object for $n > 1$. T_1 refers to the first arrival time.). In this work, all T_n are considered to be independent and identically

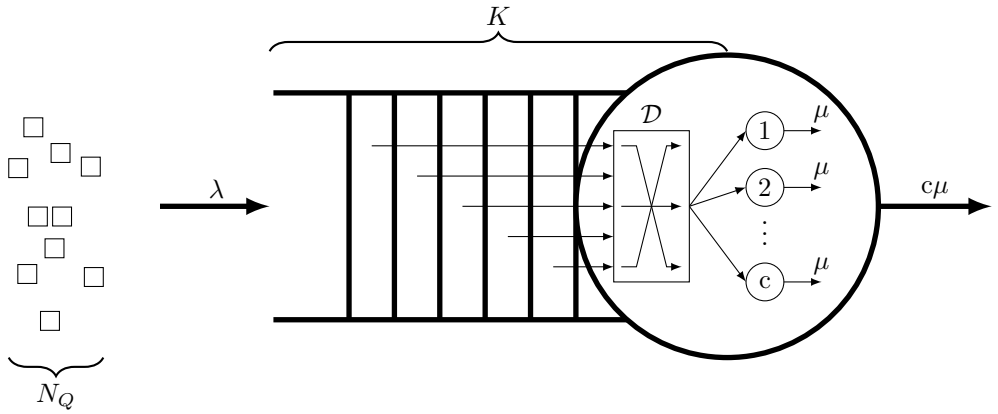


Fig. 2.4. A queuing system. Objects from a population of size N_Q arrive according to an inter-arrival time with distribution A (rate λ) and will be queued, if the overall capacity K is not exceeded. A queuing discipline $\mathcal{D}$ decides how the objects are being scheduled to the c servers. Each server has a service time with distribution B (rate μ).

distributed (i.i.d.) and so to have all the same distribution A. The mean value of the arrival rate is denoted by $\lambda := \mathbb{E}\left[T^{-1}\right]$.

Service time distribution. The service time S_n is an RV that describes the time which is needed to process the n^{th} object. Here, all S_n are assumed to be i.i.d. from one distribution B. Thus, each process has the same mean service rate $\mu := \mathbb{E}\left[S^{-1}\right]$.

Number of servers. The number of servers c defines how many objects can be processed in parallel. In a real scenario this can refer to the number of independent cores, slots, or RBs.

Capacity The capacity K is the maximum number of objects a queue can hold, which includes the objects currently being in service. Whenever a queue reached its capacity, further arrivals will be dropped, which can be referred to as admission control. If it is not specified, $K = \infty$.

Population The population N_Q determines the number of existing objects that can arrive. Usually $N_Q = \infty$ is assumed.

Queuing Discipline The queuing discipline $\mathcal{D}$ determines how the objects are being served. Common examples are first in first out (FIFO) or last in first out (LIFO), where objects are scheduled one after the other, or processor sharing (PS) with multiple objects being served at once. Here, the default is FIFO.

Let the important i.i.d. property of the arrival and service process be fixed in the following assumption throughout this thesis.

Assumption 2.1 (Independent Arrival and Service)**.** *The inter-arrival times at one queuing system are i.i.d.. The same property holds for the service times at one queue.*

From both, the arrival and the service process, the important property of *network load* $\rho = \frac{\lambda}{\mu}$ can be derived. A queuing system without admission control (i. e., $K = \infty$) can only be *stable* if $\rho \leq \mathrm{c}$ holds, because otherwise more objects arrive than the system is able to handle. Depending on the concrete stochastic processes, in most cases even the strict inequality $\rho < \mathrm{c}$ is required.[6]

For the sake of notation, *Kendall's notation* has been established [Ken53]. By listing all properties in the template $A/B/\mathrm{c}/K/N_Q/\mathcal{D}$ the type of the queue is fully described. Thereby, starting from the right end, default properties are usually left out. For instance, the prevalent $\mathrm{M}/\mathrm{M}/1$ model is short for $\mathrm{M}/\mathrm{M}/1/\infty/\infty/\mathrm{FIFO}$. Here, the M denotes a Markovian, i. e., a memoryless, process.

In this work, the most important models comprise $\mathrm{M}/\mathrm{M}/1$, $\mathrm{M}/\mathrm{D}/1$ (deterministic service, e. g., fixed packet size), and the general $\mathrm{GI}/\mathrm{GI}/1$ (general independent arrival and service) queues.

Over time, a lot of queuing models have been studied, and analytical expressions for the first stochastic moments (mean and variance) or even the PDF of waiting and sojourn time have been derived depending on the analytical tractability. However, in the context of URLLC, the very high percentiles of the distribution, i. e., the tail, are of significant importance. In contrast, mean values exhibit only little value of information. A main contribution of this thesis is the proposal of a numerical algorithm to obtain the waiting time distribution for the very general $\mathrm{GI}/\mathrm{GI}/1$ queues (cf. Section 4.3). This way, also high percentiles can be evaluated in practically relevant situations.

2.2.2 Application to Wireless Systems

Since the air interface is a shared medium and user behavior is usually a random process, queuing theory appears to be a suitable tool for the performance evaluation of wireless systems. Consequently, it has gained a large attraction with

[6] An example where $\rho = \mathrm{c}$ guarantees stability is given by the deterministic system of a $\mathrm{D}/\mathrm{D}/\mathrm{c}$ queue. In contrast, $\mathrm{M}/\mathrm{M}/\mathrm{c}$ models require $\rho < \mathrm{c}$ to be stable.

regard to performance evaluation. In [Bon05] flow-level models were used to increase performance by exploiting fading effects through opportunistic scheduling. Further, Bonald et al. studied the impact of mobility and derived tight performance bounds in [BBP04], and in [Bon04b] they also started to incorporate the impact of inter-cell interference (ICI) by deriving second-order performance approximations, as well as bounds based on full or no interference.

In [Kle+16], a powerful performance evaluation framework to analyze the DL of interference-coupled cellular networks has been introduced, which not only enabled to study, e. g., capacity optimization in heterogeneous networks (HetNets) [KGF14], admission control [KFF14], or video performance [KF15], but also provides the foundation for the work described in Chapter 3. There, extensions in terms of the considered dynamics as well as approaches for the UL are explained.

2.3 Queuing Networks

Instead of considering only single queues, $M \in \mathbb{N}$ queues $Q_1 \dots Q_M$ can be connected to build up a network, which is referred to as a *queuing network* and illustrated in Fig. 2.5. Queuing networks were introduced already in the mid of the 20$^{\text{th}}$ century. For instance, the simplest model, where only $\mathrm{M}/\mathrm{M}/1$ queues build the network, is referred to a *Jackson network* [Jac57]. There exist also more sophisticated models, e. g., *Kelly networks* [Kel75] or *BCMP networks* [Bas+75], which even support different classes of customers. For both types, equilibrium state probabilities can be derived under certain conditions. However, since the assumptions of all mentioned models are too specific and do not provide expressions for the waiting time distributions, this thesis follows a different approach. In the subsequent section, the general model is introduced.

2.3.1 Definitions and Basic Concept

The set of the queue indices is denoted by $\mathcal{M} := \{1, \dots, M\}$. Each of the single queues can be characterized then as described in Section 2.2.1. Objects arriving with a rate α to the entire system will start at queue j according to the probability p_{0j}. Whenever an object is processed at queue i, it will be forwarded to queue j with probability p_{ij} or eventually leave the system with probability p_{i0}. The vectors $\boldsymbol{p}_{0\cdot} = (p_{0j})$ and $\boldsymbol{p}_{\cdot 0} = (p_{i0})$ contain the arrival departure probabilities,

respectively. The *routing matrix* $\boldsymbol{P} := (p_{ij})$ collects the routing probabilities and the *extended routing matrix* $\tilde{\boldsymbol{P}}$ accommodates all values

$$\tilde{\boldsymbol{P}} := \left(\begin{array}{c|c} —\boldsymbol{p}_{0\cdot}^T— & 0 \\ \hline \boldsymbol{P} & \begin{array}{c}|\\ \boldsymbol{p}_{\cdot 0}\\ |\end{array} \end{array}\right). \tag{2.1}$$

Since the entries are probabilities, the following equation must hold

$$\tilde{\boldsymbol{P}} \cdot \mathbf{1} = \mathbf{1}, \tag{2.2}$$

where $\mathbf{1} = (1, \ldots, 1)^T$ denotes a column vector of suitable dimension. In addition, in every queuing network the individual arrival rate λ_i of queue Q_i can be obtained from the following conservation law:

$$\forall j \in \mathcal{M}: \qquad \lambda_j = \alpha p_{0j} + \sum_{i \in \mathcal{M}} p_{ij}\lambda_i \tag{2.3}$$

or equivalently

$$(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{\lambda} = \alpha \boldsymbol{p}_{0\cdot}, \tag{2.4}$$

which balances the overall and individual arrival rates α and λ_i, respectively.

The matrix $(\boldsymbol{I} - \boldsymbol{P})$ is regular for non-generated[7] queuing networks and so the system 2.4 has a unique solution $\boldsymbol{\lambda}$. For a regularity proof, the reader is referred to [BG92, Section 3.8].

With regard to latency, the sojourn time J_i of queue Q_i is of major interest. It is the RV of the time an object spends in the queue and comprises the RVs waiting time W_i and service time S_i as follows[8]

$$J_i = W_i + S_i. \tag{2.5}$$

[7] A non-generated queuing network refers to a network, where each object eventually leaves the system with probability 1. This means in particular that the network must not contain any loops, which cannot be left.

[8] For the sake of notation, a small inaccuracy was introduced here with respect to the indices. Whereas in the previous explanation S_n referred to the service time of the n^{th} object of one queue, the index i refers now to the (limiting, stable) service time of queue Q_i (i. e., $S_i := \lim_{n\to\infty} S_{i,n}$). The author decided to forgo introducing any complicated separated notation and clearly states instead what is meant and uses the indices n and i to differentiate. The same holds also for other RVs, such as the waiting and sojourn time.

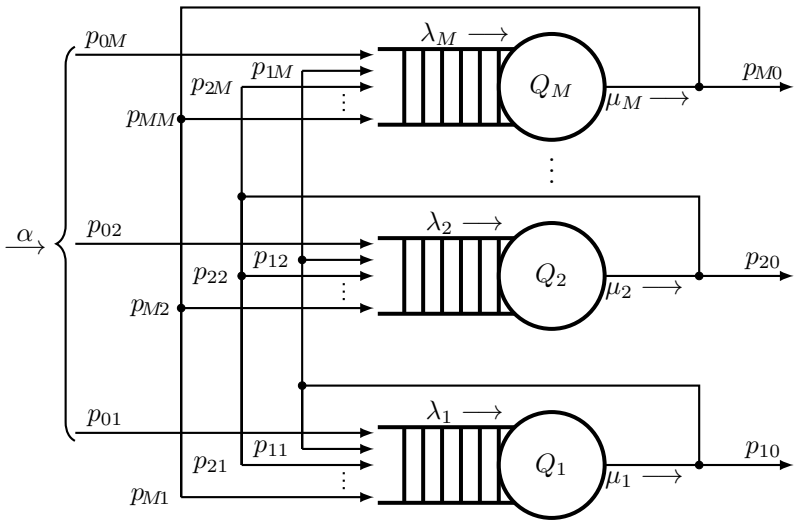


Fig. 2.5. A general queuing network. Queues are connected with each other and forward processed objects to another queue or out of the system according to the routing probabilities p_{ij}.

Since the waiting time W_i only depends on previous arrival and service times and due to Assumption 2.1, both addends are independent, Eq. (2.5) can also be expressed as the convolution of the respective PDFs.

$$f_{J_i} = f_{W_i} * f_{S_i}. \tag{2.6}$$

For the E2E latency analysis, conducted later (4.2), the overall sojourn time along a path through the queuing network should be considered. Therefore, let $\boldsymbol{q} := (q_1, \ldots, q_\kappa) \in \mathcal{M}^\kappa$ define the indices of such a path. Clearly, the overall sojourn time $J_{\boldsymbol{q}}$, waiting time $W_{\boldsymbol{q}}$, and service time $S_{\boldsymbol{q}}$ along the path $\boldsymbol{q}$ are the sum of the individual times at each queue

$$J_{\boldsymbol{q}} = \sum_{i=1}^{\kappa} J_{q_i}, \qquad W_{\boldsymbol{q}} = \sum_{i=1}^{\kappa} W_{q_i}, \qquad S_{\boldsymbol{q}} = \sum_{i=1}^{\kappa} S_{q_i}. \tag{2.7}$$

Unfortunately, the addends Eqs. (2.7) are not mutually independent in this case, because the waiting in one queue may correlate with the one of a subsequent queue. However, the analysis of dependent queuing systems is hardly tractable and only few research on very specific situations exists in this area (e. g., [Cal81]). Therefore, in this thesis, the following assumption is made.

Assumption 2.2 (Independent Waiting at Different Queues)**.** *The waiting times W_i at the queues Q_i, $i \in \mathcal{M}$, are assumed to be independent from each other.*

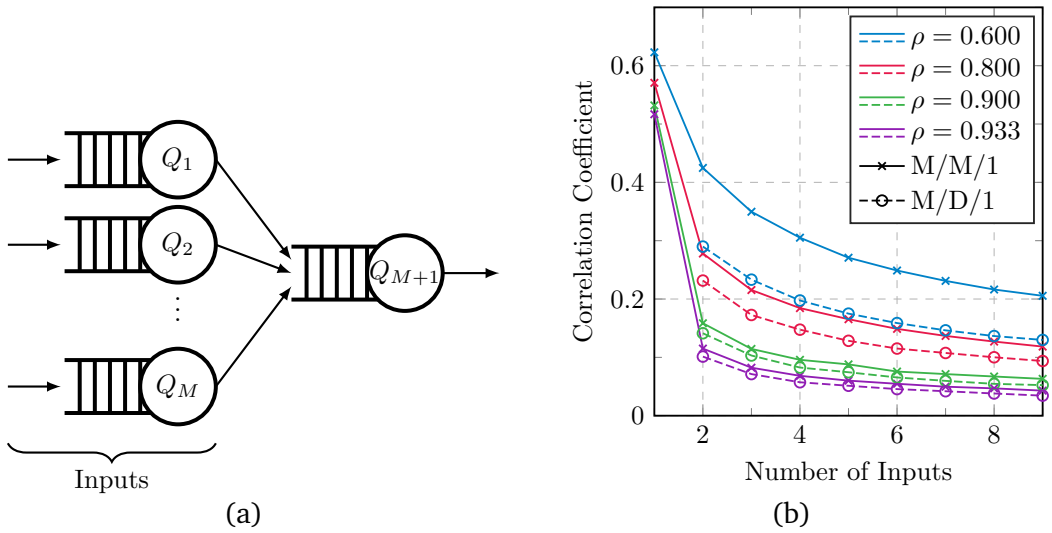


Fig. 2.6. Illustration of the Kleinrock independence approximation. (a) The queuing network that was used to study the impact of the network density, i. e., number of inputs, on the correlation of sojourn times. (b) Correlation coefficient between the sojourn time of one of the input queues and the output queue.

On first glance, Assumption 2.2 appears to be very restrictive. However, the *Kleinrock Independence Approximation* (c.f. Sec. 3.6.1 in [BG92]) states that this assumption results in a good approximation, as long as the considered network is dense enough, because then the randomness of other streams decreases the relative contribution of a particular stream.

This principle is illustrated in Fig. 2.6. A simple tree-shaped queuing network (Fig. 2.6a) with a varying number M of queues was studied in a simulation. Here, $(M-1)$ queues serve as an input for one output queue Q_M. In this setting the correlation between the waiting time W_1 in one of the input queues Q_1 and the waiting time W_M in the output queue Q_M was measured. The results for $\mathrm{M/M/1}$ and $\mathrm{M/D/1}$[9] queues are depicted in Fig. 2.6b, respectively. It can be observed that the correlation decreases for a higher number of input queues (network density) and for a higher network load ρ. Especially for high loads the correlation vanishes fast, which is of particular interest as these are the interesting situations for worst case analyses.

[9] If two identical $\mathrm{M/D/1}$ queues are in tandem, the waiting time in the second queue is always zero, because each object is processed in the same service time. When an object moves forward to the next queue, its predecessor leaves the second queue in the same moment. Thus, there are no valid data points in the plot for one input in this case.

Based on Assumption 2.2, the PDF of the waiting time $W_{\boldsymbol{q}}$ can be formulated as a convolution

$$f_{W_{\boldsymbol{q}}} = f_{W_{q_1}} * f_{W_{q_2}} * \ldots * f_{W_{q_\kappa}} =: \overset{\kappa}{\underset{i=1}{\circledast}} f_{W_{q_i}}. \tag{2.8}$$

In contrast, the service times S_{q_i} along path $\boldsymbol{q}$ are assumed to be highly mutually dependent. No matter on which time scale the analysis is conducted, let it be a flow or a packet, the service time will depend on the object size (i. e., flow size or packet length) at each visited node. To incorporate this, it is assumed that for each object an initial object size S_0 is drawn. All service times S_i of this object at an arbitrary queue Q_i will be a scaled version, based on the respective service rate μ_i

$$S_i = \mu_i^{-1} S_0, \tag{2.9}$$

which leads to the accumulated service time along path $\boldsymbol{q}$

$$S_{\boldsymbol{q}} = \left(\sum_{i=1}^{\kappa} \mu_{q_i}^{-1} \right) S_0 =: \mu_{\boldsymbol{q}}^{-1} S_0, \tag{2.10}$$

and translates to the PDF

$$f_{S_{\boldsymbol{q}}}(t) = \mu_{\boldsymbol{q}} f_{S_0}\left(\mu_{\boldsymbol{q}} t\right). \tag{2.11}$$

With this and Eq. (2.8), the PDF of the overall sojourn time $J_{\boldsymbol{q}}$ along path $\boldsymbol{q}$ can be expressed as

$$f_{J_{\boldsymbol{q}}}(t) = \left[\left(\overset{\kappa}{\underset{i=1}{\circledast}} f_{W_{q_i}}(s) \right) * \mu_{\boldsymbol{q}} f_{S_0}\left(\mu_{\boldsymbol{q}} s\right) \right](t). \tag{2.12}$$

Here, the inner argument s was introduced to differentiate between the argument and the integration variable of the involved convolutions. The advantage here is that with the PDF, also the cumulative distribution function (CDF) is provided by integrating Eq. (2.12) as follows[10]

$$F_{J_{\boldsymbol{q}}}(t) = \int_{-\infty}^{t} f_{J_{\boldsymbol{q}}}(\tau) \mathrm{d}\tau = \left[\left(\overset{\kappa}{\underset{i=1}{\circledast}} f_{W_{q_i}}(s) \right) * F_{S_0}\left(\mu_{\boldsymbol{q}} s\right) \right](t). \tag{2.13}$$

[10] Here, the simple integration of two convoluted functions $\int f * g = f * G$ was exploited, which suits best for integrating the service time term. Of course, any of the f_{W_i} could have been integrated as well, leading to a more complicated expression.

The CDF F_{J_q} then naturally provides the r^{th} percentile by evaluating $F_{J_q}^{-1}\left(\frac{r}{100}\right)$, what is much more valuable for URLLC than having only statements about mean and variance.

Details to this approach can be found in [Sch+19c]. Furthermore, a concrete application is presented in Chapter 4.

2.3.2 Application to Radio Access Networks

As an extension to separated queuing systems, queuing networks gained a lot attention for studying wireless systems as well. Kühn did extensive research on queuing theory and queuing networks in the context of computing and communications. As early as 1976, he introduced a general framework [Küh76] for the approximate analysis of complex queuing networks by decomposition. The framework can handle general queuing stations. Even though it is limited to the first two moments, i. e., mean and variance, of the input and output processes, it can provide valuable insights into system performances. This fundamental work can also be applied to current and future technologies, as proposed in his more recent work about 5G system architecture [Küh15], incorporating also SDNs and NFV.

Especially in the field of SDN, queuing networks can provide valuable insights for their optimal configuration and so for exploiting their potential. In this regard, Jarschel et al. studied the performance of an SDN controller and SDN switch with the help of a two-node queuing network in [Jar+11; Mah+14]. The authors took the model parameters from real measurements and compared analytical results to simulations. In [Mah+15], they even extended their model to an SDN network with multiple nodes. Their work is based on Jackson networks [Jac57], tailored to their investigated situation, and therefore also limited to the assumption of exponential inter-arrival and service. Wireless SDN have also been approximately analyzed with the help of queuing networks in [ZNS14] with respect to queuing delay and loss, where the theory of large deviations has been applied, and the results have been compared to simulations based on realistic traffic models from [NGM08a].

How queuing networks have been applied in the scope of this thesis, is described in Chapter 4.

Tab. 2.2. Extract of the most important implemented functionality of the distribution class.

Distribution Class	
Method	Description
$+, -, \cdot$	distribution of the sum/difference/product of two independent RVs
$<, \leq, >, \geq$	probability of the comparison of two independent RVs, i. e., "$X < Y$" $:= \mathbb{P}[X < Y]$
$\min, \max$	distribution of the minimum/maximum of two (or more) RVs
$aX, X + a$	distribution of the scaled/shifted RV by a scalar a
$\mathbb{E}[\cdot], \mathbb{E}[f(\cdot)]$	expectation of an RV or of a function of an RV
$\mathrm{median}(\cdot)$	median of an RV
$\mathrm{percentile}(\cdot, q)$	qth percentile of an RV
$d(X, Y)$	distance between RVs X and Y in different norms, with respect to there PDF or CDF
$\mathrm{sample}(\cdot, N)$	draw N samples from RV
$\mathrm{DistFromSamples}(\cdot)$	approximate a distribution from given samples

2.4 Distribution Class

A fundamental part of this thesis and the modeling is the new MATLAB [Mat19] class implemented by the author within this context that holds stochastic distributions. The idea is that such a class simplifies the handling of all the involved RVs. For instance, the sum of two independent RVs could be written in the code as simple as $Z = X + Y$, but is internally treated as the convolution $f_Z = f_X * f_Y$ of the respective PDFs. Whenever the analytical solution of such an expression is known (and implemented), the class will make use of it. If, for instance, $X, Y \sim \mathrm{Exp}(\lambda)$ in the same example, then $Z \sim \Gamma(2, \lambda)$. Otherwise, an operation can always be performed numerically.

An object of the class can be initiated as one of several standard distributions or with an arbitrary PDF or probability mass function (PMF), provided as numerical values. In particular, the class can handle continuous, discrete, and mixtures of both kinds of distributions, which is in particular helpful with respect to waiting times. A full description of the implemented class is out of the scope of this thesis. Instead, a short overview of the functionality is provided in Table 2.2. Here, it should be noted that the focus is rather on the functionalities, which were important with respect to the thesis, than on implementing a comprehensive framework. The class is published online at the MATLAB central file exchange [Sch19].

Even though there already exists a powerful distribution class in MATLAB, the already existing one is more tailored to other purposes, such as distribution fitting, sampling, and statistical analysis. Thus, the aforementioned required functionality is not available.

2.5 Summary

This chapter builds the foundation for the subsequent work. It provides an overview about existing traffic models for 5G use cases that is the starting point for realistic performance evaluations. Thereby, the modeling considers not only the temporal, but also the spatial domain, which has particular importance for performance evaluation in the presence of traffic hot spots. Especially for mMTC and cMTC, still a great research potential in finding sophisticated models remains. Furthermore, two sources of empirical data, namely social networks and crowd-sourced app data, are presented.

Moreover, basic concepts of queuing theory and queuing networks, which form the mathematical framework used in this thesis, are explained. With this, important notation aspects are introduced, basic assumptions are set, and the foundation for the E2E latency modeling along a path in a network is led. In queuing-theoretic terms latency refers to sojourn time, which is related to the waiting time. For both aspects, individual (potentially coupled) queues and queuing networks, a short overview about state of the art applications of this theory to wireless communications is provided. Finally, the implementation of a class for handling stochastic distributions is presented, which has a major role in the model evaluation within this thesis.

3

Modeling the Wireless Access

After a theoretical foundation was built in the previous chapter, the model is now tailored to a cellular scenario with respect to DL traffic. The general system model used in this chapter mainly builds on the framework introduced in [Kle+16]. After introducing the concrete setup, the model's potential will be shown by evaluating a concrete eMBB application, i. e., video streaming, in Section 3.2. The major contribution here, compared with previous work, is the incorporation of the complex interactions of ICI dynamics, leading to a system of multiple multi-class PS queues with mutually coupled service rates. Since the arising systems of (differential) equations are not trivial to be solved, particular attention is spend on this issue (cf. Section 3.2.4). Furthermore, potential steps for a model extension to the UL will be depicted. The chapter also includes a comparison of model and simulation in regard to computational effort and complexity (Section 3.4). To facilitate reading, Table 3.1 provides an overview about the essential notation used in this chapter.

3.1 Wireless Downlink Access

Let the scenario, which is illustrated in Fig. 3.1, consist of a finite number $N \in \mathbb{N}_{>0}$ of BSs, which are deployed in a region $\mathcal{L} \in \mathbb{R}^2$. Their index set is denoted as $\mathcal{N} := \{1, \ldots, N\}$. For the performance analysis, each BS i is modeled as a queue Q_i. In this region, (user) positions are depicted as $\boldsymbol{u} \in \mathcal{L}$. The location of a user follows its PDF $f_{\boldsymbol{u}}(\boldsymbol{u})$, which allows for heterogeneous spatial user distributions (cf. Section 2.1.2). It is assumed that each possible user location $\boldsymbol{u}$ is associated uniquely with one BS. Thus, each BS i is associated with a cell $\mathcal{L}_i$ for $i \in \mathcal{N}$ and the terms cell, BS, and queue are used interchangeable. Thereby, the cells $\mathcal{L}_i$ form a *partition*[1] of $\mathcal{L}$. The users are assumed to arrive according to

[1]The family of sets $\{\mathcal{L}_i\}$ is called a *partition* of $\mathcal{L}$, iff for all $i, j \in \mathcal{N}$ the following three properties hold: (i) $\mathcal{L}_i \neq \emptyset$, (ii) $\mathcal{L}_i \cap \mathcal{L}_j = \emptyset$ for $i \neq j$, and (iii) $\bigcup_{k \in \mathcal{N}} \mathcal{L}_k = \mathcal{L}$.

Tab. 3.1. Overview about the essential notation in Chapter 3. Indices are omitted for readability.

Variable	Space	Description
t, h	$\mathbb{R}$	Time and infinitesimal time difference
q, Q	$\mathbb{R}$	Buffer variable and buffer process
i	$\mathcal{N}$	Cell index
$\boldsymbol{u}$	$\mathcal{L}$	User position
States		
$\boldsymbol{x}, \boldsymbol{X}$	$\mathcal{X}$	BS state
z	$\mathcal{Z}^{\mathrm{PF}}, \mathcal{Z}^{\mathrm{PB}}$	Observed state by the tagged flow (during prefetching and playback)
$\boldsymbol{y}, \boldsymbol{Y}$	$\mathcal{Y}$	Interference scenario (realization and RV)
Rates		
c	$[0, c_{\max}]$	Achievable rate based on SINR γ
λ, μ	$\mathbb{R}_{>0}$	Arrival and service rate of a BS
ϑ	$\mathbb{R}_{>0}$	Observed departure rate of other flows
μ'	$\mathbb{R}_{>0}$	Observed deactivation rate of other BSs
ψ	$\mathbb{R}_{>0}$	Rate to the absorbing state $\mathbf{A}$
$\Psi, \tilde{\Psi}$	$\mathbb{R}_{>0}$	Sum of all rates leaving a state (with and without absorbing state $\mathbf{A}$)
v, w	$\mathbb{R}_{>0}$	Download rate and effective download rate
$\boldsymbol{A}, \boldsymbol{B}$	$\mathbb{R}^{LK \times LK}$	Diagonal matrices collecting v, w
$\boldsymbol{M}$	$\mathbb{R}^{LK \times LK}$	Matrix collecting state transition rates
$\boldsymbol{\Lambda}$	$\mathbb{R}^{K \times K}$	Diagonal matrix of transition rates between interference scenarios
Probabilities		
π	$[0, 1]$	State probability
$\boldsymbol{\Pi}$	$[0, 1]$	Joint probability of observed competing flows and interference
ζ	$[0, 1]$	Probability of observing an interference scenario after admission
$P^{\mathrm{st}}, P^{\mathrm{b}}$	$[0, 1]$	Starvation and blocking probability
Variables of the PDE and ODE systems		
$\boldsymbol{U}$	$[0, 1]^{LK}$	Startup delay distribution function
$\boldsymbol{V}$	$[0, 1]^{LK \times LK}$	Probabilities to start prefetching in one state and end in another
$\boldsymbol{W}$	$[0, 1]^{LK}$	Starvation probability function

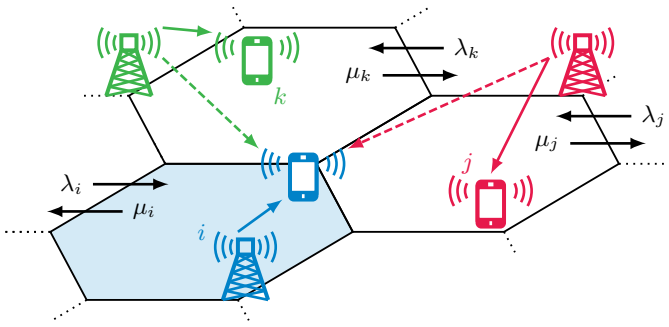


Fig. 3.1. The dynamics in a cellular network for the DL. The performance of a UE is affected by the dynamically changing load conditions in its own cell as well as the randomly varying interference (dashed arrows) from neighboring cells.

a Poisson process with constant rate[2] λ to the overall scenario. With appropriate weighting, the individual arrival rate λ_i of cell i and the conditioned PDF $f_{\boldsymbol{u},i}$ of users in cell i can be derived as

$$\lambda_i = \lambda \int_{\mathcal{L}_i} f_{\boldsymbol{u}}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}, \qquad \text{and} \tag{3.1}$$

$$f_{\boldsymbol{u},i}(\boldsymbol{u}) = \frac{f_{\boldsymbol{u}}(\boldsymbol{u})}{\int_{\mathcal{L}_i} f_{\boldsymbol{u}}(\boldsymbol{v}) \mathrm{d}\boldsymbol{v}}. \tag{3.2}$$

For this chapter, an exponentially distributed file size with mean Ω is assumed. More details on the traffic model will be provided for the concrete application in Section 3.2.

Within this regime, the performance a user is impacted by the (randomly changing) numbers of users at the same BS, but also by the experienced interference from other BSs, which again randomly changes with user activity in neighboring cells. Related to this, the following assumption is stated.

Assumption 3.1 (Best effort service)**.** *Whenever a BS serves at least one user, it is said to be* active *and* inactive *otherwise. An active BS is assumed to allocate all of its radio resources to the users in service (best effort) and, thus, transmits with full power.*

Based on this assumption, let the binary variable y_i denote the activity of the i^{th} BS, i. e., $y_i = 1$ if BS i is active and zero otherwise. All these components are collected in the joint interference state vector $\boldsymbol{y} := (y_i) \in \mathcal{Y} := \{0, 1\}^N$. The

[2]Here, a stationary process is assumed. In general, λ can also be time-dependent as in the more general model formulation in [Kle+16]. However the steady state analysis requires a stationary process.

set $\mathcal{Y}$ is referred to as the interference state space. With this, it is convenient to partition the set of cell indices into the ones of inactive and active BSs with respect to the interference scenario $\boldsymbol{y}$ as follows

$$\mathcal{N}_0(\boldsymbol{y}) := \{i \in \mathcal{N} \mid y_i = 0\} \qquad \text{and} \qquad \mathcal{N}_1(\boldsymbol{y}) := \{i \in \mathcal{N} \mid y_i = 1\}\,. \tag{3.3}$$

The active BSs determine the interference a user experiences. With $p_j^{\mathrm{rx}}(\boldsymbol{u})$ being the received power of a UE at position $\boldsymbol{u}$ from BS j for any $j \in \mathcal{N}$, which follows the path loss model from [3GP10]

$$l_{\mathrm{path}}(d_j) = 128.1\,\mathrm{dB} + 37.6 \log_{10}\left(\frac{d_j}{\mathrm{km}}\right)\mathrm{dB} \tag{3.4}$$

for the distance d_j from the user to the BS j, the signal-to-interference-plus-noise ratio (SINR) γ_i of a user associated with BS i can be expressed as

$$\gamma_i(\boldsymbol{u}, \boldsymbol{y}) := \frac{p_i^{\mathrm{rx}}(\boldsymbol{u})}{\sum_{j \in \mathcal{N}_1(\boldsymbol{y}) \setminus \{i\}} p_j^{\mathrm{rx}}(\boldsymbol{u}) + B_{\mathrm{BW}} N_0}. \tag{3.5}$$

Here, B_{BW} and N_0 denote the bandwidth and the spectral density of the thermal noise power, respectively. The SINR can be mapped to an achievable rate c_i

$$c_i(\boldsymbol{u}, \boldsymbol{y}) := e_{\mathrm{BW}} B_{\mathrm{BW}} \min\left\{\log_2\left(1 + e_{\mathrm{SINR}} \gamma_i(\boldsymbol{u}, \boldsymbol{y})\right), c_{\max}\right\}, \tag{3.6}$$

by a modification of the theoretical capacity bound provided by Shannon's law [Sha49] to a more realistic bound incorporating the bandwidth efficiency e_{BW}, the SINR efficiency e_{SINR} and a maximum achievable data rate $c_{\max}$ of the considered wireless system [Mog+07]. Following the approach of [Bon05], the average service rate μ_i of BS i for a fixed interference scenario $\boldsymbol{y}$ results from the user-density-weighted harmonic mean over the cell i

$$\mu_i(\boldsymbol{y}) := \frac{1}{\Omega}\left[\int_{\mathcal{L}_i} \frac{f_{\boldsymbol{u}}(\boldsymbol{u})}{c_i(\boldsymbol{u}, \boldsymbol{y})} \mathrm{d}\boldsymbol{u}\right]^{-1}. \tag{3.7}$$

Since the interference scenario $\boldsymbol{y}$ dynamically changes with the activity of other BSs, it becomes apparent that the service rate is not constant. Thus, the system at hand constitutes an interference-coupled system of queues, which will be discussed in the Sections 3.1.1 and 3.1.2.

At each BS, a resource-fair scheduler is assumed. This can be accomplished by the round robin (RR) scheduler which translates to the PS discipline in

queuing terminology. To avoid congestion, each BS applies admission control. A simple yet accurate way to incorporate this into the queuing model is setting the capacity of queue Q_i to a finite $K_i \in \mathbb{N}_{>0}$ leading to a stable queuing system.

3.1.1 Steady State Performance

To analyze the dynamics, the random process of the state $\boldsymbol{X}_i(t)$ (with realization x_i) of queue Q_i is considered. Together, they form the multivariate process $\boldsymbol{X}(t) = (\boldsymbol{X}_i(t)) \in \mathcal{X} := \{0, \dots, K_1\} \times \cdots \times \{0, \dots, K_M\}$ (with realization $\boldsymbol{x}$). This process generates the random process of interference scenarios $\boldsymbol{Y}(t) = (Y_i) := \operatorname{sgn}(\boldsymbol{X}(t)) \in \mathcal{Y}$ with their already introduced realizations $\boldsymbol{y}$.

A first step of the performance evaluation is constituted by the derivation of the joint steady-state distribution $\pi(\boldsymbol{x}) := \mathbb{P}[\boldsymbol{X}(t) = \boldsymbol{x}]$. With the aforementioned assumptions, a system of multi-class PS queuing systems with mutually interference-modulated service rates is defined, which is barely tractable. However, several approximation methods have been proposed:

(i) first and second order (approximated) performance bounds based on a full interference (first order, lower), no interference (first order, upper), fluid (second order, lower), and a quasi-stationary (second order, upper bound) regime, respectively, in [Bon04b],

(ii) average interference performance approximation [SY12], and

(iii) aggregation of variables performance approximation [FF12].

A comparison of all three approaches was conducted in [Kle+16]. There, approach (iii) was identified as the most accurate method, and is consequently chosen for state probability approximation in this thesis.

Key Performance Indicators

Once the steady state probabilities $\pi(\boldsymbol{x})$ are available, a series of relevant KPIs can derived. An overview is given in Table 3.2. The table also provides the respective formulas for the standard queuing models $\mathrm{M/M/1/\infty}$ and $\mathrm{M/M/1}/K$ with PS as the scheduling discipline.

Tab. 3.2. KPIs for an $\mathrm{M/M/1/\infty}$ PS queue Q_i, an $\mathrm{M/M/1}/K_i$ PS queue Q_i, or a general queuing system Q_i with admission control and known state probabilities $\pi(\boldsymbol{x})$.

KPI	Symbol	$\mathrm{M/M/1/\infty}$ PS	$\mathrm{M/M/1}/\boldsymbol{K_i}$ PS	General system with admission control
Resource utilization	η_i	ρ_i	$\rho_i \left(1 - P_i^{\mathrm{b}}\right)$	$\sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ x_i>0}} \pi(\boldsymbol{x})$
Blocking probability	P_i^{b}	0	$\frac{(1-\rho_i)\,\rho_i^{K_i}}{1-\rho_i^{K_i+1}}$	$\sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ x_i=K_i}} \pi(\boldsymbol{x})$
Mean number of active flows	$\bar{n}_i$	$\frac{\rho_i}{1-\rho_i}$	$\frac{\rho_i}{1-\rho_i} - \frac{(K_i+1)\rho_i^{K_i+1}}{1-\rho_i^{K_i+1}}$	$\sum_{x=1}^{K_i} x \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ x_i=x}} \pi(\boldsymbol{x})$
Mean sojourn time	$\mathbb{E}[J_i]$	$\frac{1}{\mu_i-\lambda_i}$	$\frac{\bar{n}_i}{\lambda_i\left(1-P_i^{\mathrm{b}}\right)}$	
Cell throughput	$R_{\mathrm{cell},i}$	$\frac{\lambda_i\Omega}{\rho_i} = \Omega\mu_i$	$\frac{\lambda_i\left(1-P^{\mathrm{b}}\right)\Omega}{\rho_i} = \left(1-P^{\mathrm{b}}\right)\Omega\mu_i$	
Flow throughput	$R_{\mathrm{flow},i}$	$R_{\mathrm{cell},i}\,(1-\rho_i)$	$\frac{\eta_i R_{\mathrm{cell},i}}{\bar{n}_i}$	
Reference		[Kle75]	[KFF14]	[Kle+16]

3.1.2 The Network observed by a Tagged Flow

In the following, the perspective of the analysis changes from a steady state investigation to the transient behavior of the entire system. For this purpose, a "tagged flow" that arrives at BS i is considered, a perspective that was already used in [Xu+13] for the derivation of streaming KPIs. This tagged flow is admitted to the service of its associated BS, whenever the BS is not at full load, due to admission control. The admission probability is the complement of the blocking probability, i. e., $1 - P_i^{\mathrm{b}}$, which is provided by Table 3.2. After admission the tagged flow experiences the interference scenario $\boldsymbol{y}$ according to the conditioned probability

$$\zeta_i(\boldsymbol{y}) = \mathbb{P}\left[\boldsymbol{Y}(t) = \boldsymbol{y} \mid \boldsymbol{X}_i(t) \geq 1\right] = \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ \mathrm{sgn}(\boldsymbol{x})=\boldsymbol{y} \\ x_i\geq 1}} \pi(\boldsymbol{x}) \Bigg/ \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ x_i\geq 1}} \pi(\boldsymbol{x}), \tag{3.8}$$

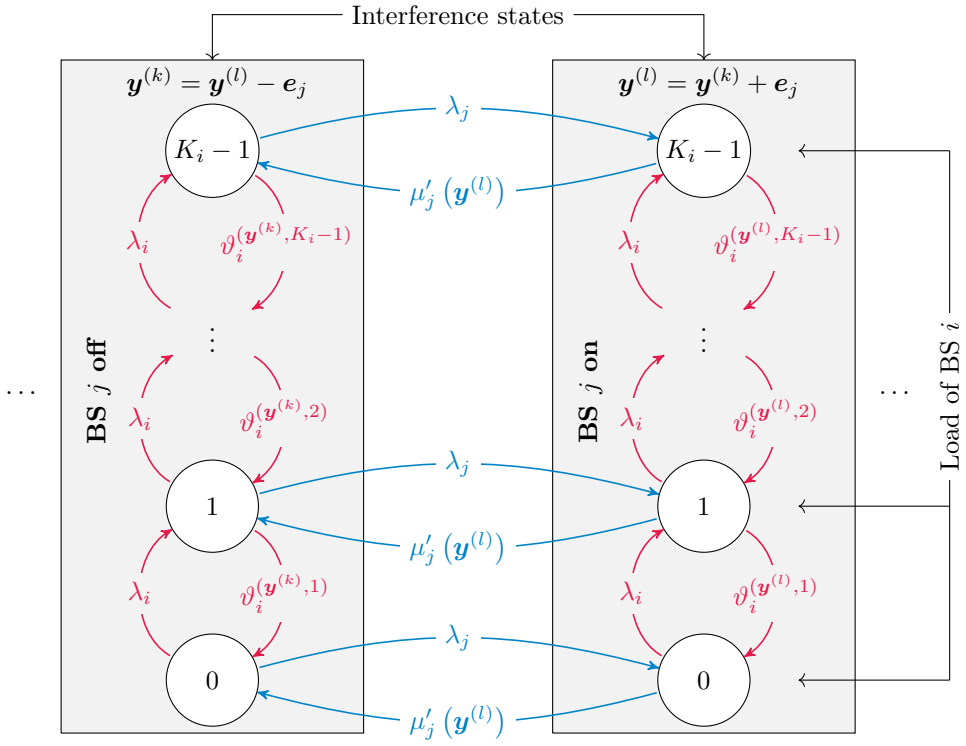


Fig. 3.2. Extract of the Markov model for the network dynamics as observed by a tagged flow in cell i. The illustration shows only two of the N dimensions in total. Two adjacent interference scenarios $\boldsymbol{y}^{(k)}$ and $\boldsymbol{y}^{(l)}$, which differ in the j^{th} component, are depicted as rectangular boxes ($j \neq i$). Transitions between the boxes (blue) stem from activation and deactivation of BS j, i. e., events (E_c) and (E_d). Transitions referring to other BSs are not shown. The transitions within a box (red) are due to changing cell load caused by arrivals or departures of competing flows (events (E_a) and (E_b)).

and upon arrival the tagged flow is exposed to a number z_i of other competing flows in the same cell according to the conditioned probability

$$\pi_{z_i} = \mathbb{P}\left[\boldsymbol{X}_i(t) = z_i \mid \boldsymbol{X}_i(t) < K_i\right] = \frac{1}{1 - P_i^{\text{b}}} \sum_{\substack{\boldsymbol{x} \in \mathcal{X} \\ x_i = z_i}} \pi(\boldsymbol{x}). \tag{3.9}$$

Due to the system dynamics, this number may change according to an auxiliary random process $Z_i^{\text{PF}}(t) \in \mathcal{Z}_i^{\text{PF}} := \{0, \ldots, K_i - 1\}$. As long as the flow is in service, $\boldsymbol{X}_i = Z_i^{\text{PF}} + 1$ holds for the cell load of BS i.

The cell load of the considered BS i may change due to the following *inner* events at the same BS

(E_a) arrival of a new flow with rate λ_i, or

(E_b) departure of *another* finished flow with rate $\vartheta_i^{(\boldsymbol{y},z_i)} := \frac{z_i}{z_i+1}\mu_i(\boldsymbol{y})$,

based on the current interference scenario $\boldsymbol{y}$ and the number of currently observed competing flows z_i.

While the tagged flow is active, the interference process is restricted to scenarios where the i^{th} BS is active. This modified process is denoted as $\tilde{\boldsymbol{Y}}(t) \in \mathcal{Y}_i := \{\boldsymbol{y} \in \mathcal{Y} | y_i = 1\}$ with a modified interference state space $\mathcal{Y}_i$ with cardinality $L := |\mathcal{Y}_i| = 2^{N-1}$. The interference changes according to the following possible *outer* events at other BSs

(E_c) activation of neighboring BS j with rate λ_j, or

(E_d) deactivation of a BS j caused by the departure of its last served flow with the conditioned rate $\mu'_j(\boldsymbol{y}) := \mu_j(\boldsymbol{y}) \cdot \mathbb{P}[x_j = 1 | x_j \geq 1]$.

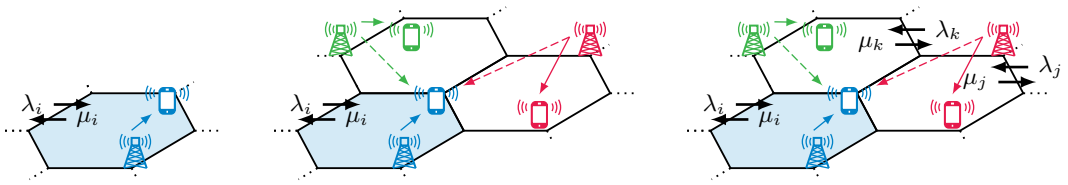
On a continuous timescale, it is reasonable to state the following assumption.

Assumption 3.2. *Only one of the four events (E_a)... (E_d) may happen at exactly the same time instance.*

Considering all events (E_a)–(E_d), a Markov process is formed as illustrated in Fig. 3.2. The states $(\boldsymbol{y}, z_i)$ comprise the interference scenario with $N-1$ degrees of freedom (DoFs) and the observed number of competing flows with one DoF, and so the model has N dimensions in total. Of course, the figure can only show two of the dimensions, i. e., two adjacent interference scenarios $\boldsymbol{y}^{(k)} = \boldsymbol{y}^{(l)} - \boldsymbol{e}_j$ for suitable $k, l \in \mathcal{N}$, which are depicted as rectangular boxes. The illustration shows the inner transitions due to the events (E_a) and (E_b) within one interference scenario, as well as the transitions between two scenarios due to the events (E_c) and (E_d). The process will be the foundation for the video streaming model, which is derived in the subsequent section.

3.2 Application to Streaming Traffic

According to the Cisco Visual Networking Index [Cis17], video streaming will account for 78 % of an estimated total global mobile traffic demand of 49 exabytes per month by 2021. This fact alone makes video streaming one of the most important use cases within eMBB. In this regard, not only legacy quality of service (QoS) metrics, but also quality of experience (QoE) metrics are of significant importance, especially from an operator's or video provider's perspective, who are both interested in satisfied clients. Indeed, studies [KS12] demonstrated



(a) Single BS (b) Quasi-static interference (c) Dynamic interference

Fig. 3.3. Comparison of the major differences between the presented approaches. (a) [Xu+13; Xu+16] considers a single BS only, (b) [KF15] extends the work by quasi-stationary interference, (c) [Sch+20b; SKF17] and this thesis also include interference dynamics.

that users already start abandoning videos, if they have to wait longer than two seconds for buffering.

In this regard, two video streaming KPIs are of interest, namely the *startup delay distribution* as well as the *buffer starvation probability*, which are related to latency and reliability, respectively. The derivation of those KPIs follows an initial idea of Xu et al., e. g., in [Xu+13; Xu+16]. The authors modeled competing flows at a single BS as depicted in Fig. 3.3a. This approach was extended in [KF15] by introducing interference in neighbor cells (Fig. 3.3b). However, interference was only incorporated in a quasi-stationary regime. In our work [Sch+20b; SKF17], the following extensions were contributed:

(i) The KPI buffer starvation probability is derived in the multi-cellular interference coupled context.

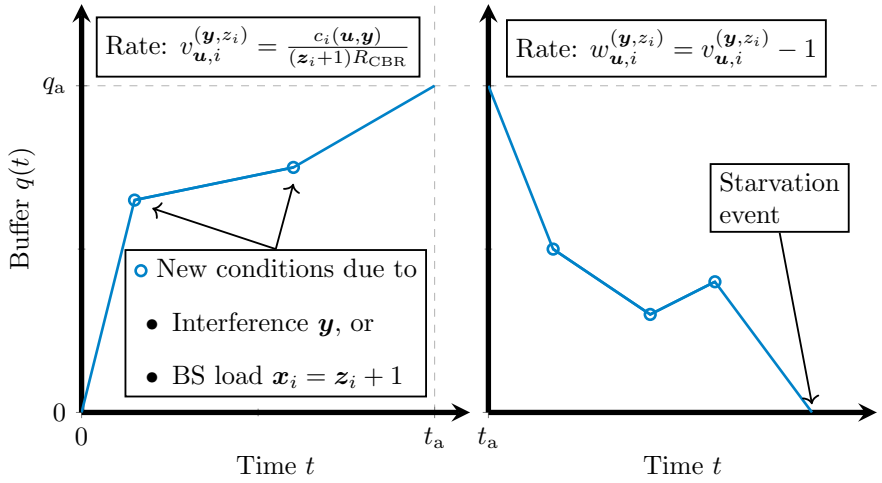
(ii) The models for both the startup delay distribution and the buffer starvation probability are extended by incorporating also interference dynamics as illustrated in Fig. 3.3c (cf. Sections 3.2.1 and 3.2.3).

(iii) Three different approximation methods for solving the involved PDE systems are proposed, which, in contrast to [Xu+13], do not suffer from numerical instabilities, such as oscillations. (Details are provided in Section 3.2.4)

(iv) An extensive study of the impact of different system parameters, namely admission control, the video bitrate, and traffic demand in neighboring cells, on the video streaming KPIs was conducted in [Sch+20b].

(v) The models were validated through extensive system-level simulations.

(vi) A unified QoE metric was introduced in [SKF17].



(a) Prefetching (b) Playback

Fig. 3.4. Phases of Video Streaming. (a) During prefetching video content is downloaded at varying data rates until a certain threshold q_a is reached at the startup delay time t_a. (b) During playback the buffer is simultaneously filled and emptied.

Characterization of the Video Traffic

The authors of [Xu+16; Xu+17] introduce approaches for modeling variable bitrate (VBR) and hyper-exponentially distributed video lengths, respectively. However, for this study, the length of a video T_video is assumed to be exponentially distributed with mean $\bar{T}_\mathrm{video}$ as this already reveals essential features of video traffic according to [Xu+13] (cf. Section 2.1.1). Furthermore, videos with a constant bitrate (CBR) of R_CBR are studied. Even though CBR is not as efficient as the more sophisticated VBR encoding, CBR is still widely-used, especially in the field of live streaming, where simple encoding and bitrates without variations are favored. The reason for choosing this traffic model is that the contribution and focus here are rather on the cellular network dynamics than on the video streaming. In principle, the model can be extended by applying the approaches of Xu et al. as well. Bringing both together, i. e., studying the interference dynamics with more sophisticated traffic models, is left for further studies.

Video Streaming Phases

As depicted in Fig. 3.4, two phases of video streaming are considered. First, there is a *prefetching* phase (Fig. 3.4a), in which video content will be buffered, to reduce the risk of buffer starvation. Within this phase, the video buffer is filled with a rate $v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} = \frac{c_i(\boldsymbol{u},\boldsymbol{y})}{(z_i+1)R_\mathrm{CBR}}$ (measured in seconds of video content per second), depending on the current state $(\boldsymbol{y}, z_i)$ and the user location $\boldsymbol{u}$. After

a predefined threshold of video content q_a (measured in seconds) is buffered, the second phase, i. e., the *playback*, starts (Fig. 3.4b). The time when the prefetching finishes and the playback starts is called the startup delay t_a. During the playback, the buffer will be simultaneously filled with the download rate and emptied with a rate of one second of video content per second, which leads to an effective buffer change rate $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} = v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} - 1$. Since the effective rate can also be negative, this may lead eventually to an empty buffer and, thus, to playback interruption.

3.2.1 Derivation of the Startup Delay Distribution

In this section, the distribution of the startup delay t_a will be derived. With the distribution of an RV, all relevant stochastic measures, such as mean, variance, or percentiles, will also be provided.

For this purpose, the change of the video buffer content $Q(t)$ over time will be studied. As proposed in [Xu+13], the process of emptying the content buffer at the sending BS will be considered, which constitutes the dual problem of looking into the filling of the buffer at the UE. This approach simplifies the analysis. From this perspective, the buffer changes with rate $v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$ according to $Q(t) = Q(t-h) - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h$ in an infinitesimal interval $[t-h, t]$.

With this, let $U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q)$ define the probability that the tagged flow from a user at location $\boldsymbol{u}$ and observing the state $(\boldsymbol{y}, z_i)$ will finish downloading the remaining video content q in a time t or less. For this quantity, the following balance equations[3] for the transition from $(t-h, q - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h)$ to (t,q) can be formulated by incorporating the possible transitions of the events (E_a)–(E_d) in the lines (3.10b)–(3.10e), respectively

$$
\begin{aligned}
U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q) = {} & \left(1 - \Psi_i^{(\boldsymbol{y},z_i)} h\right) U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} \left(t-h, q - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h\right) && \text{(3.10a)} \\
& + \lambda_i h U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i+1)} \left(t-h, q - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h\right) && \text{(3.10b)} \\
& + \vartheta_i^{(\boldsymbol{y},z_i)} h U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i-1)} \left(t-h, q - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h\right) && \text{(3.10c)} \\
& + \sum_{j \in \mathcal{N}_0(\boldsymbol{y})} \lambda_j h U_{\boldsymbol{u},i}^{(\boldsymbol{y}+\boldsymbol{e}_j,z_i)} \left(t-h, q - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h\right) && \text{(3.10d)} \\
& + \underbrace{\sum_{j \in \mathcal{N}_1(\boldsymbol{y}) \setminus \{i\}} \mu_j'(\boldsymbol{y}) h}_{\text{rate}} \underbrace{U_{\boldsymbol{u},i}^{(\boldsymbol{y}-\boldsymbol{e}_j,z_i)}}_{\text{state}} \underbrace{\left(t-h, q - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h\right)}_{\text{previous time and buffer}} && \text{(3.10e)}
\end{aligned}
$$

[3] It should be noted that Eq. (3.10) is the general form. At the boundaries, i. e., for $z_i \in \{0, K_i\}$, such that $(\boldsymbol{y}, z_i - 1)$ or $(\boldsymbol{y}, z_i + 1)$ are not valid, the respective terms are left out.

with $\Psi_i^{(\boldsymbol{y},z_i)}$ being the sum of all transition rates from state $(\boldsymbol{y}, z_i)$ to other states

$$\Psi_i^{(\boldsymbol{y},z_i)} := \lambda_i + \vartheta_i^{(\boldsymbol{y},z_i)} + \sum_{j\in\mathcal{N}_0(\boldsymbol{y})} \lambda_j \; + \sum_{j\in\mathcal{N}_1(\boldsymbol{y})\setminus\{i\}} \mu_j'(\boldsymbol{y}), \tag{3.10f}$$

such that $\left(1-\Psi_i^{(\boldsymbol{y},z_i)}h\right)$ is the fraction that stays in state $U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$ in line (3.10a). By subtracting $U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}\left(t-h, q-v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}h\right)$ in (3.10), dividing by h and determining the limit $h\to 0$, the following system of coupled PDEs for every possible state $(\boldsymbol{y}, z_i)$ is derived

$$\begin{aligned}
&\frac{\partial}{\partial t}U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q) + v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}\frac{\partial}{\partial q}U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q) \\
&= -\Psi_i^{(\boldsymbol{y},z_i)}U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q) + \lambda_i U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i+1)}(t,q) + \vartheta_i^{(\boldsymbol{y},z_i)}U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i-1)}(t,q) \\
&\quad + \sum_{j\in\mathcal{N}_0(\boldsymbol{y})}\lambda_j U_{\boldsymbol{u},i}^{(\boldsymbol{y}+\boldsymbol{e}_j,z_i)}(t,q) + \sum_{j\in\mathcal{N}_1(\boldsymbol{y})\setminus\{i\}}\mu_j'(\boldsymbol{y})U_{\boldsymbol{u},i}^{(\boldsymbol{y}-\boldsymbol{e}_j,z_i)}(t,q)\,.
\end{aligned} \tag{3.11}$$

By collecting the components $U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$ of all states $(\boldsymbol{y}, z_i)$ in a vector

$$\boldsymbol{U}_{\boldsymbol{u},i} := \left[U_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(1)},0)},\ldots,U_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(1)},K_i-1)},\ldots,U_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(L)},0)},\ldots,U_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(L)},K_i-1)}\right]^T, \tag{3.12}$$

for an arbitrary numbering of the interference scenarios $\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(L)}$, the PDE system (3.11) can be formulated in a matrix-vector notation

$$\frac{\partial}{\partial t}\boldsymbol{U}_{\boldsymbol{u},i}(t,q) + \boldsymbol{A}_{\boldsymbol{u},i}\frac{\partial}{\partial q}\boldsymbol{U}_{\boldsymbol{u},i}(t,q) = \boldsymbol{M}_{\boldsymbol{u},i}\boldsymbol{U}_{\boldsymbol{u},i}(t,q). \tag{3.13}$$

Therein, the diagonal matrix $\boldsymbol{A}_{\boldsymbol{u},i}\in\mathbb{R}^{LK_i\times LK_i}$ contains the download rates $v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$ of each state $(\boldsymbol{y}, z_i)$ in its diagonal

$$\boldsymbol{A}_{\boldsymbol{u},i} := \operatorname{diag}\left(v_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(1)},0)},\ldots,v_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(1)},K_i-1)},\ldots,v_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(L)},0)},\ldots,v_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(L)},K_i-1)}\right), \tag{3.14}$$

and the matrix $\boldsymbol{M}_{\boldsymbol{u},i}$ collects all transition rates of the Markov Model

$$\boldsymbol{M}_{\boldsymbol{u},i} = \begin{bmatrix} \boldsymbol{M}_{\text{in}}(\boldsymbol{y}^{(1)}) & \boldsymbol{\Lambda}_{12} & \cdots & \boldsymbol{\Lambda}_{1L} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{M}_{\text{in}}(\boldsymbol{y}^{(2)}) & \cdots & \boldsymbol{\Lambda}_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Lambda}_{L1} & \boldsymbol{\Lambda}_{L2} & \cdots & \boldsymbol{M}_{\text{in}}(\boldsymbol{y}^{(L)}) \end{bmatrix} \in \mathbb{R}^{LK_i\times LK_i}. \tag{3.15}$$

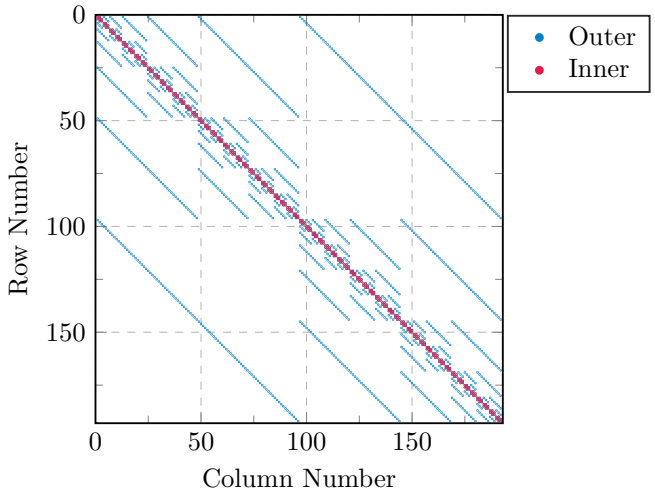


Fig. 3.5. Sparsity Pattern of $\boldsymbol{M}_{\boldsymbol{u},i}$ and $\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}$. Inner and outer transitions are indicated in red and blue, respectively.

Here, the blocks $\boldsymbol{M}_{\text{in}}(\boldsymbol{y})$ and $\boldsymbol{\Lambda}_{pq}$ are $K_i \times K_i$ submatrices and contain the inner transition rates within one interference scenario

$$\boldsymbol{M}_{\text{in}}(\boldsymbol{y}) = \begin{bmatrix} -\Psi_i^{(\boldsymbol{y},0)} & \lambda_i & & \\ \vartheta_i^{(\boldsymbol{y},1)} & -\Psi_i^{(\boldsymbol{y},1)} & \lambda_i & \\ & \ddots & \ddots & \\ & & \vartheta_i^{(\boldsymbol{y},K_i-1)} & -\Psi_i^{(\boldsymbol{y},K_i-1)} \end{bmatrix}, \tag{3.16}$$

and the outer transition rates between adjacent interference scenarios

$$\boldsymbol{\Lambda}_{mn} = \begin{cases} \lambda_j \boldsymbol{I} & \text{if } \exists j \in \mathcal{N} : \boldsymbol{y}^{(n)} = \boldsymbol{y}^{(m)} + \boldsymbol{e}_j, \\ \mu_j'(\boldsymbol{y}^{(n)}) \boldsymbol{I} & \text{if } \exists j \in \mathcal{N} : \boldsymbol{y}^{(n)} = \boldsymbol{y}^{(m)} - \boldsymbol{e}_j, \\ \boldsymbol{0}, & \text{else} \end{cases} \tag{3.17}$$

for $1 \leq m, n \leq L$, respectively. Here, $\boldsymbol{e}_i$ denotes the i^{th} unit vector. The structure of the sparse matrix $\boldsymbol{M}_{\boldsymbol{u},i}$ is illustrated in Fig. 3.5.

To complete the definition of the PDE system (3.13), initial and boundary conditions have to be added as described in the following (cf. Fig. 3.6). Since any positive content q cannot be downloaded in a time less than $t = 0$, the initial probability has to be zero (cf. Eq. (3.18a)). In contrast, zero content is downloaded in any positive time for sure, which forms the first boundary for $q = 0$ in Eq. (3.18b). For the other boundary, i. e., $q \to \infty$, and any fixed time t

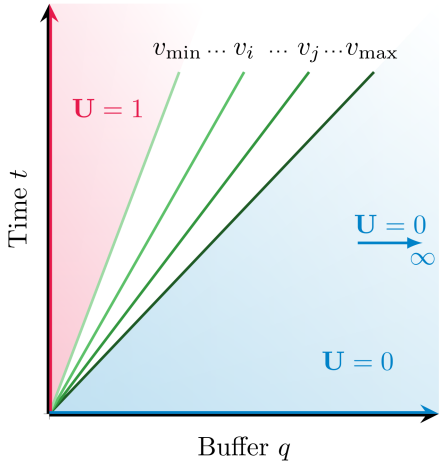


Fig. 3.6. Illustration of the discontinuous initial and boundary conditions for the startup delay distribution. For each state those will be "transported" according to the download rates $v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$. By choosing the maximum download rate, a region can be identified, where $\boldsymbol{U}_{\boldsymbol{u},i}$ is zero for all states, i. e., where the content q is too large to be downloaded within time t.

there is always a sufficiently large content q that cannot be downloaded for sure. Accordingly, the initial and boundary conditions can be formulated as follows

$$\boldsymbol{U}_{\boldsymbol{u},i}(0,q) = \boldsymbol{0} \qquad \forall q > 0, \tag{3.18a}$$

$$\boldsymbol{U}_{\boldsymbol{u},i}(t,0) = \boldsymbol{1}, \qquad \lim_{q\to\infty} \boldsymbol{U}_{\boldsymbol{u},i}(t,q) = \boldsymbol{0} \qquad \forall t \geq 0. \tag{3.18b}$$

The PDE system (3.13) together with its boundary conditions (3.18) in time and space[4] constitute a multi-dimensional system of transport equations with discontinuous boundary conditions. Determining the solution is challenging and will be discussed in Section 3.2.4.

Let the solution of the system (3.13), (3.18) now be denoted as $\boldsymbol{U}_{\boldsymbol{u},i}$. With this, the startup delay distribution $\bar{U}_{\boldsymbol{u},i}$ at a given location $\boldsymbol{u}$ can be obtained by summing over all possibly observed interference scenarios and cell loads weighted by the respective probabilities $\zeta_i(\boldsymbol{y})\pi_{z_i}$ upon arrival

$$\bar{U}_{\boldsymbol{u},i}(t,q_\mathrm{a}) = \sum_{\boldsymbol{y}\in\mathcal{Y}_i} \zeta_i(\boldsymbol{y}) \sum_{z_i=0}^{K_i-1} \pi_{z_i} U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q_\mathrm{a}). \tag{3.19}$$

[4]Transport equations usually represent physical processes, e. g., the spatio-temporal behavior of the concentration of one or multiple liquids, in time and space. This can be mapped to the situation here as follows. The different states refer to different liquids. Their probabilities are transported in the same way as the concentration of the different liquids, but with the buffer q being the spatial variable. Finally, the conservation law, given by the PDE, guarantees that the sum of probability or matter, respectively, remains constant.

Finally, a weighted integration over all cell locations leads to the startup delay distribution $\bar{U}_i$ in the entire cell

$$\bar{U}_i(t, q_\mathrm{a}) = \int_{\mathcal{L}_i} \bar{U}_{\boldsymbol{u},i}(t, q_\mathrm{a}) f_{\boldsymbol{u},i}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}. \tag{3.20}$$

3.2.2 Transient State Probabilities

Before the starvation probability can be derived, the focus has to be put on how the state probabilities have changed during prefetching. Accordingly, the random process $\boldsymbol{\mathcal{I}}(t) \in \{(\boldsymbol{y}, z_i) | \boldsymbol{y} \in \mathcal{Y}_i, z_i \in \mathcal{Z}_i^{\mathrm{PF}}\}$ of the observed states while buffering is considered. Let

$$V_{(\boldsymbol{y},z_i)}^{(\boldsymbol{\xi},\eta_i)}(q; q_\mathrm{a}) := \mathbb{P}\left[\boldsymbol{\mathcal{I}}(t_\mathrm{a}) = (\boldsymbol{\xi}, \eta_i) | \boldsymbol{\mathcal{I}}(t_0) = (\boldsymbol{y}, z_i), Q(t_0) = q\right] \tag{3.21}$$

define the probability[5] of ending the prefetching in state $(\boldsymbol{\xi}, \eta_i)$ conditioned on having content q already prefetched and being in state $(\boldsymbol{y}, z_i)$ at a time t_0. This time, the perspective is on the primary process of filling the buffer at the UE according to

$$Q(t) = Q(t-h) + v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h \tag{3.22}$$

within an infinitesimal interval $[t, t+h]$. Again, as with the derivation of Eq. (3.10), balance equations can be formulated for all combinations of $(\boldsymbol{y}, z_i), (\boldsymbol{\xi}, \eta_i)$ as follows

$$\begin{aligned} V_{(\boldsymbol{y},z_i)}^{(\boldsymbol{\xi},\eta_i)}(q; q_\mathrm{a}) = {} & \left(1 - \Psi_i^{(\boldsymbol{y},z_i)} h\right) V_{(\boldsymbol{y},z_i)}^{(\boldsymbol{\xi},\eta_i)} \left(q + v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h; q_\mathrm{a}\right) \\ & + \lambda_i h V_{(\boldsymbol{y},z_i+1)}^{(\boldsymbol{\xi},\eta_i)} \left(q + v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h; q_\mathrm{a}\right) \\ & + \vartheta_i^{(\boldsymbol{y},z_i)} h V_{(\boldsymbol{y},z_i-1)}^{(\boldsymbol{\xi},\eta_i)} \left(q + v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h; q_\mathrm{a}\right) \\ & + \sum_{j \in \mathcal{N}_0(\boldsymbol{y})} \lambda_j h V_{(\boldsymbol{y}+\boldsymbol{e}_j,z_i)}^{(\boldsymbol{\xi},\eta_i)} \left(q + v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h; q_\mathrm{a}\right) \\ & + \sum_{j \in \mathcal{N}_1(\boldsymbol{y}) \setminus \{i\}} \mu_j'(\boldsymbol{y}) h V_{(\boldsymbol{y}-\boldsymbol{e}_j,z_i)}^{(\boldsymbol{\xi},\eta_i)} \left(q + v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h; q_\mathrm{a}\right). \end{aligned} \tag{3.23}$$

This time the dependence on t is eliminated through Eq. (3.22), since, in contrast to the previous section, the time is not of interest here. By repeating the same

[5] It should be noted that $V_{(\boldsymbol{y},z_i)}^{(\boldsymbol{\xi},\eta_i)}$ also depends on the location $\boldsymbol{u}$ and the cell i. However, for the sake of notation, the respective indices have been left out in this section.

steps as with the balance equations (3.10) and collecting the possible state transitions $V_{(\boldsymbol{y},z_i)}^{(\boldsymbol{\xi},\eta_i)}$ in a matrix $\boldsymbol{V}_{\boldsymbol{u},i}$, as follows

$$\boldsymbol{V}_{\boldsymbol{u},i} := \begin{bmatrix} V_{(\boldsymbol{y}^{(1)},0)}^{(\boldsymbol{y}^{(1)},0)} & \cdots & V_{(\boldsymbol{y}^{(1)},0)}^{(\boldsymbol{y}^{(1)},K_i)} & V_{(\boldsymbol{y}^{(1)},0)}^{(\boldsymbol{y}^{(2)},0)} & \cdots \\ \vdots & & \vdots & \vdots & \\ V_{(\boldsymbol{y}^{(1)},K_i)}^{(\boldsymbol{y}^{(1)},0)} & \cdots & V_{(\boldsymbol{y}^{(1)},K_i)}^{(\boldsymbol{y}^{(1)},K_i)} & V_{(\boldsymbol{y}^{(1)},K_i)}^{(\boldsymbol{y}^{(2)},0)} & \cdots \\ V_{(\boldsymbol{y}^{(2)},0)}^{(\boldsymbol{y}^{(1)},0)} & \cdots & V_{(\boldsymbol{y}^{(2)},0)}^{(\boldsymbol{y}^{(1)},K_i)} & V_{(\boldsymbol{y}^{(2)},0)}^{(\boldsymbol{y}^{(2)},0)} & \cdots \\ \vdots & & \vdots & \vdots & \ddots \end{bmatrix}, \tag{3.24}$$

a coupled system of ordinary differential equations (ODEs) can be derived as

$$\boldsymbol{A}_{\boldsymbol{u},i} \frac{\mathrm{d}}{\mathrm{d}q} \boldsymbol{V}_{\boldsymbol{u},i}(q; q_\mathrm{a}) = -\boldsymbol{M}_{\boldsymbol{u},i} \boldsymbol{V}_{\boldsymbol{u},i}(q; q_\mathrm{a}). \tag{3.25}$$

Since $\boldsymbol{A}_{\boldsymbol{u},i}$ is diagonal with positive entries, its inverse exists and the auxiliary matrix $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{V}} := \boldsymbol{A}_{\boldsymbol{u},i}^{-1} \boldsymbol{M}_{\boldsymbol{u},i}$ is well defined. Thus, Eq. (3.25) becomes

$$\frac{\mathrm{d}}{\mathrm{d}q} \boldsymbol{V}_{\boldsymbol{u},i}(q; q_\mathrm{a}) = -\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{V}} \boldsymbol{V}_{\boldsymbol{u},i}(q; q_\mathrm{a}). \tag{3.26}$$

This linear ODE system has the well-known solution

$$\boldsymbol{V}_{\boldsymbol{u},i}(q; q_\mathrm{a}) = \exp(-\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{V}} q) \boldsymbol{V}_{\boldsymbol{u},i}(0; q_\mathrm{a}) \tag{3.27}$$

with $\exp(\cdot)$ being the matrix exponential. To derive the initial value $\boldsymbol{V}_{\boldsymbol{u},i}(0; q_\mathrm{a})$, the bounding condition for $q = q_\mathrm{a}$ is considered, for which the starting and finishing state are identical for sure

$$\boldsymbol{V}_{\boldsymbol{u},i}(q_\mathrm{a}; q_\mathrm{a}) = \boldsymbol{I}. \tag{3.28}$$

Inserting Eq. (3.28) now into (3.27) for $q = q_\mathrm{a}$ leads to the initial condition

$$\boldsymbol{V}_{\boldsymbol{u},i}(0; q_\mathrm{a}) = \exp(\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{V}} q_\mathrm{a}) \boldsymbol{V}_{\boldsymbol{u},i}(q_\mathrm{a}; q_\mathrm{a}) = \exp(\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{V}} q_\mathrm{a}) \tag{3.29}$$

and, thus, to the explicit solution

$$\boldsymbol{V}_{\boldsymbol{u},i}(q; q_\mathrm{a}) = \exp(-\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{V}} q) \exp(\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{V}} q_\mathrm{a}) = \exp\left(\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{V}} (q_\mathrm{a} - q)\right). \tag{3.30}$$

3.2.3 Derivation of the Starvation Probability

For the derivation of the buffer starvation probability, the playback phase has to be studied. Here, it is convenient to introduce an *absorbing state* $\mathbf{A}$ as proposed in [Xu+13]. The absorbing state captures the event that the tagged flow finishes the playback and cannot suffer from starvation anymore. Because flow sizes are exponentially distributed and PS is assumed, the rate to the absorbing state becomes

$$\psi_i^{(\boldsymbol{y},z_i)} = \frac{\mu_i(\boldsymbol{y})}{z_i+1} \tag{3.31}$$

and so the following relation holds

$$\vartheta_i^{(\boldsymbol{y},z_i)} + \psi_i^{(\boldsymbol{y},z_i)} = \mu_i(\boldsymbol{y}). \tag{3.32}$$

With the absorbing state, the process of observed competing flows is slightly modified to $Z^{\mathrm{PB}}(t) \in \mathcal{Z}^{\mathrm{PB}} := \{0,\dots,K_i-1,\mathbf{A}\}$. In addition, the video buffer now changes with the effective rate $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$ during an infinitesimal interval $[t,t+h]$

$$Q(t+h) = Q(t) + w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} h. \tag{3.33}$$

Similar to the derivation of Eq. (3.10), balance equations can be formulated for the probability $W^{(\boldsymbol{y},z_i)}(q)$ that a flow at location $\boldsymbol{u}$, currently observing state $(\boldsymbol{y},z_i)$ and having content q buffered, will suffer from buffer starvation, by incorporating the events (E_a)–(E_d) in lines (3.34b)–(3.34e), respectively

$$\begin{aligned}
W^{(\boldsymbol{y},z_i)}(q) = {} & \left(1-\tilde{\Psi}_i^{(\boldsymbol{y},z_i)}h\right)W^{(\boldsymbol{y},z_i)} \quad \left(q + w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}h\right) && \text{(3.34a)}\\
& + \lambda_i h W^{(\boldsymbol{y},z_i+1)} \quad \left(q + w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}h\right) && \text{(3.34b)}\\
& + \vartheta_i^{(\boldsymbol{y},z_i)} h W^{(\boldsymbol{y},z_i-1)} \quad \left(q + w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}h\right) && \text{(3.34c)}\\
& + \sum_{j\in\mathcal{N}_0(\boldsymbol{y})} \lambda_j h W^{(\boldsymbol{y}+\boldsymbol{e}_j,z_i)} \left(q + w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}h\right) && \text{(3.34d)}\\
& + \sum_{j\in\mathcal{N}_1(\boldsymbol{y})\setminus\{i\}} \mu_j'(\boldsymbol{y}) h W^{(\boldsymbol{y}-\boldsymbol{e}_j,z_i)} \left(q + w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}h\right), && \text{(3.34e)}
\end{aligned}$$

with a modification of $\Psi_i^{(\boldsymbol{y},z_i)}$ due to the new possible transition to $\mathbf{A}$ and incorporating Eq. (3.32)

$$\tilde{\Psi}_i^{(\boldsymbol{y},z_i)} := \Psi_i^{(\boldsymbol{y},z_i)} + \psi_i^{(\boldsymbol{y},z_i)} = \lambda_i + \mu_i(\boldsymbol{y}) + \sum_{j\in\mathcal{N}_0(\boldsymbol{y})} \lambda_j + \sum_{j\in\mathcal{N}_1(\boldsymbol{y})\setminus\{i\}} \mu_j'(\boldsymbol{y}). \tag{3.35}$$

Since $W^{(\boldsymbol{y},z_i)}(q) = 0$ for the state $\mathbf{A}$, the absorbing state is not visible in the balance equation, but has only influence on $\tilde{\Psi}_i^{(\boldsymbol{y},z_i)}$. Repeating the same procedure as in the previous sections leads to the ODE system

$$\boldsymbol{B}_{\boldsymbol{u},i}\frac{\mathrm{d}}{\mathrm{d}q}\boldsymbol{W}_{\boldsymbol{u},i}(q) = -\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}\boldsymbol{W}_{\boldsymbol{u},i}(q) \tag{3.36}$$

with

$$\boldsymbol{W}_{\boldsymbol{u},i} := \left[\boldsymbol{W}_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(1)},0)},\ldots,\boldsymbol{W}_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(1)},K_i-1)},\ldots,\boldsymbol{W}_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(L)},0)},\ldots,\boldsymbol{W}_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(L)},K_i-1)}\right]^T, \tag{3.37}$$

$$\boldsymbol{B}_{\boldsymbol{u},i} := \operatorname{diag}\left(w_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(1)},0)},\ldots,w_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(1)},K_i-1)},\ldots,w_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(L)},0)},\ldots,w_{\boldsymbol{u},i}^{(\boldsymbol{y}^{(L)},K_i-1)}\right). \tag{3.38}$$

and $\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}$ is a modified form of $\boldsymbol{M}_{\boldsymbol{u},i}$, where $\Psi_i^{(\boldsymbol{y},z_i)}$ is substituted by $\tilde{\Psi}_i^{(\boldsymbol{y},z_i)}$.

In contrast to the previous section, generating initial or boundary conditions requires a little more effort in this case. If there was any state $(\boldsymbol{y}, z_i)$ with an effective rate $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} = 0$, the derivative in the respective line in the system (3.36) would vanish, and so this single line would be an algebraic equation and not an ODE anymore. This algebraic notation could then be used to express the affected variable with the help of other variables, eliminate the line from the system, and solve the remaining PDE system.

Thus, without loss of generality, $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} \neq 0$ is assumed for all states $(\boldsymbol{y}, z_i)$ from now. Therefore, $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}} = \boldsymbol{B}_{\boldsymbol{u},i}^{-1}\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}$ is well-defined and the PDE system can also be formulated as

$$\frac{\mathrm{d}}{\mathrm{d}q}\boldsymbol{W}_{\boldsymbol{u},i}(q) = -\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}}\boldsymbol{W}_{\boldsymbol{u},i}(q), \tag{3.39}$$

Let the states be separated by the sign of $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$, i. e.,

$$\boldsymbol{W}_{\boldsymbol{u},i}^{-} = \left\{\boldsymbol{W}_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}\middle| w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} < 0\right\}, \qquad \boldsymbol{W}_{\boldsymbol{u},i}^{+} = \left\{\boldsymbol{W}_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}\middle| w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} > 0\right\}, \tag{3.40}$$

and let their cardinalities be denoted as $n^- = |\boldsymbol{W}_{\boldsymbol{u},i}^{-}|$ and $n^+ = |\boldsymbol{W}_{\boldsymbol{u},i}^{+}|$, respectively. Then boundary conditions for Eq. (3.39) can be stated as follows

$$\boldsymbol{W}_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(0) = 1 \qquad \text{for all } (\boldsymbol{y}, z_i) \text{ with } w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} < 0, \tag{3.41}$$

$$\lim_{q\to\infty}\boldsymbol{W}_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(q) = 0 \qquad \text{for all } (\boldsymbol{y}, z_i) \text{ with } w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} \geq 0. \tag{3.42}$$

For the states $(\boldsymbol{y}, z_i)$ with a negative effective rate $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} < 0$ starvation happens for sure, and so $\boldsymbol{W}_{\boldsymbol{u},i}^{-}(0) = \mathbf{1}$, which is formulated in Eq. (3.41) and directly provides the first n^- initial conditions. The second condition Eq. (3.42) reflects

the fact that the starvation probability is zero for a sufficiently filled video buffer. The remaining n^+ initial conditions can be obtained through the following lemma.

Lemma 3.1. *To satisfy Eq.* (3.42)*, the initial values* $\boldsymbol{W}_{\boldsymbol{u},i}(0)$ *have to meet*

$$\left(\boldsymbol{V}_M^{-1}\boldsymbol{W}_{\boldsymbol{u},i}(0)\right)^+ = 0. \tag{3.43}$$

Together with Eq. (3.41)*, the initial conditions for* $\boldsymbol{W}_{\boldsymbol{u},i}(0)$ *are uniquely defined.*

Proof. The proof is divided into two steps.

(1) For the first step it is assumed that $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}}$ is diagonalizable, i. e., $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}} = \boldsymbol{V}_M\boldsymbol{D}_M\boldsymbol{V}_M^{-1}$, where $\boldsymbol{D}_M$ is a diagonal matrix containing the eigenvalues of $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}}$. With this, the explicit solution of (3.39) can be reformulated as

$$\begin{aligned}\boldsymbol{W}_{\boldsymbol{u},i}(q) &= \exp(-\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}}q)\boldsymbol{W}_{\boldsymbol{u},i}(0), &(3.44)\\ &= \boldsymbol{V}_M\exp(-\boldsymbol{D}_M q)\boldsymbol{V}_M^{-1}\boldsymbol{W}_{\boldsymbol{u},i}(0). &(3.45)\end{aligned}$$

The matrix $\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}$ has the entries $-\tilde{\Psi}_i^{(\boldsymbol{y},z_i)} < 0$ on the diagonal. Thus, $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}} = \boldsymbol{B}_{\boldsymbol{u},i}^{-1}\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}$ has positive diagonal entries for all states $(\boldsymbol{y}, z_i)$ with $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} < 0$ and negative entries for each state $(\boldsymbol{y}, z_i)$ with $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} > 0$. Furthermore, $\tilde{\boldsymbol{M}}_{\boldsymbol{u},i} =: (m_{ij})$ is strictly diagonally dominant, since it holds that (cf. Eq. (3.35))

$$\sum_i |m_{ii}| = |\tilde{\Psi}_i^{(\boldsymbol{y},z_i)}| = |\Psi_i^{(\boldsymbol{y},z_i)} + \psi_i^{(\boldsymbol{y},z_i)}| > |\Psi_i^{(\boldsymbol{y},z_i)}| = \sum_{j\neq i}|m_{ij}|. \tag{3.46}$$

The matrix $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}} = (m_{ii}^{\boldsymbol{W}})$ inherits this property, because the diagonal matrix $\boldsymbol{B}_{\boldsymbol{u},i}^{-1}$ only scales the rows of $\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}$.

Now Gershgorin circles [Ger31, Proposition III] are applied. For each diagonal entry of $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}}$ a Gershgorin circle $\mathcal{G}_i := \bar{\mathcal{B}}\left(m_{ii}^{\boldsymbol{W}}, \sum_{j\neq i}|m_{ij}^{\boldsymbol{W}}|\right)$ is defined, with $\bar{\mathcal{B}}(x, r)$ being the closed circle around x with radius r in the complex plane. Due to the diagonal dominance, the Gershgorin circles corresponding to positive (negative) diagonal elements lie completely in the positive (negative) plane and, thus, the circles do not intersect with the ones in the negative (positive) plane. This implies one eigenvalue with

a positive real part for each $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} < 0$ and one with a negative real part for each $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} > 0$. The entries in

$$\exp(-\boldsymbol{D}_{\boldsymbol{M}} q) = \operatorname{diag}\left(e^{-(\boldsymbol{D}_{\boldsymbol{M}})_{ii} q}\right), \tag{3.47}$$

in Eq. (3.45) which correspond to the eigenvalues with a negative real part tend to ∞ as $q \to \infty$. Thus, the corresponding entries of $\boldsymbol{V}_{\boldsymbol{M}}^{-1} \boldsymbol{W}_{\boldsymbol{u},i}(0)$ have to be zero, i. e., Eq. (3.43) has to be satisified to still ensure (3.42). This provides n^+ additional linear independent constraints to determine the missing values $\boldsymbol{W}_{\boldsymbol{u},i}^{+}(0)$, since $\boldsymbol{V}_{\boldsymbol{M}}^{-1}$ is a regular matrix.

(2) If $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}}$ is not diagonalizable, similar steps can be performed with the Jordan decomposition $\boldsymbol{M}_{\boldsymbol{u},i}^{\boldsymbol{W}} = \boldsymbol{P}_{\boldsymbol{M}} \boldsymbol{J}_{\boldsymbol{M}} \boldsymbol{P}_{\boldsymbol{M}}^{-1}$ which always exists. In this case, the arguments on the diagonal matrix $\boldsymbol{D}_{\boldsymbol{M}}$ in Eq. (3.47) has to be modified for Jordan blocks $\boldsymbol{J}_l = \lambda_{\boldsymbol{M},l} \boldsymbol{I} + \boldsymbol{N}_l$ to the corresponding eigenvalue $\lambda_{\boldsymbol{M},l}$ with a nilpotent matrix $\boldsymbol{N}_l$, i. e., $\boldsymbol{N}_l^k = \boldsymbol{0}$ for some integer k, as follows

$$\begin{aligned}\exp(-\boldsymbol{J}_l q) &= \exp\left(-(\lambda_{\boldsymbol{M},l} q \boldsymbol{I} + q \boldsymbol{N}_l)\right) = \exp\left(-\lambda_{\boldsymbol{M},l} q \boldsymbol{I}\right) \exp\left(-q \boldsymbol{N}_l\right) \\ &= \operatorname{diag}\left(e^{-\lambda_{\boldsymbol{M},l} q}\right) \left(1 + q\boldsymbol{N} + \frac{1}{2}(q\boldsymbol{N})^2 + \cdots + \frac{1}{(k-1)!}(q\boldsymbol{N})^{k-1}\right).\end{aligned} \tag{3.48}$$

As the first factor of the right-hand side (RHS) in Eq. (3.48) is exponential and the second one is only polynomial, the same argument as in the first step can be applied, which concludes the proof.

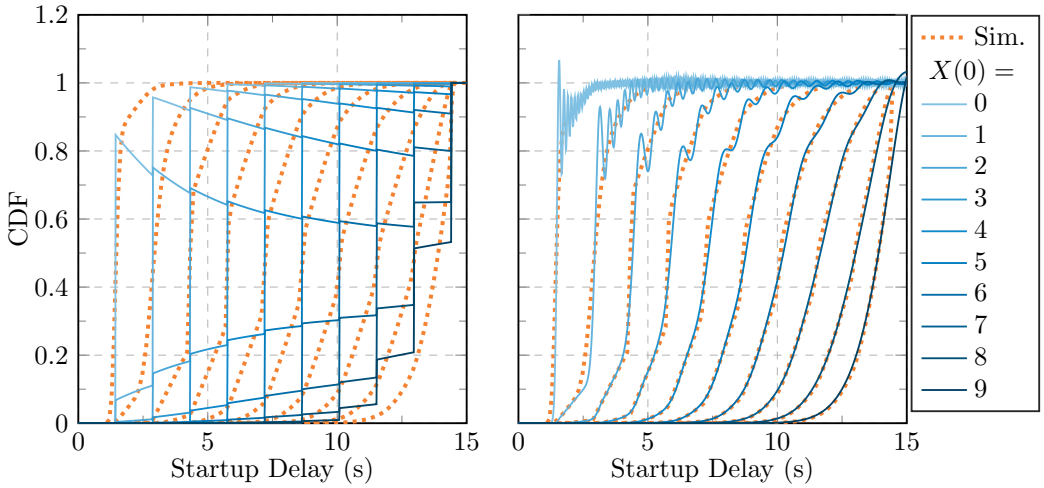
□

Finally, the starvation probability for a fixed location $\boldsymbol{u}$ can be obtained by putting everything together and including also the probability $\mathbb{P}(T_{\text{video}} > q_{\text{a}})$, that the video length exceeds the pre-buffering threshold,

$$\begin{aligned}P_{\boldsymbol{u},i}^{\text{st}}(q_{\text{a}}) &= \mathbb{P}(T_{\text{video}} > q_{\text{a}}) \cdot \boldsymbol{\Pi}^T \boldsymbol{V}_{\boldsymbol{u},i}(0; q_{\text{a}}) \boldsymbol{W}_{\boldsymbol{u},i}(q_{\text{a}}) \\ &= \exp(-q_{\text{a}}/\bar{T}_{\text{video}}) \cdot \boldsymbol{\Pi}^T \boldsymbol{V}_{\boldsymbol{u},i}(0; q_{\text{a}}) \boldsymbol{W}_{\boldsymbol{u},i}(q_{\text{a}})\end{aligned} \tag{3.49}$$

with the state probability vector

$$\boldsymbol{\Pi} = \left[\zeta_i(\boldsymbol{y}^{(1)})\pi_0, \ldots, \zeta_i(\boldsymbol{y}^{(1)})\pi_{K_i-1}, \ldots, \zeta_i(\boldsymbol{y}^{(L)})\pi_0, \ldots, \zeta_i(\boldsymbol{y}^{(L)})\pi_{K_i-1}\right]^T. \tag{3.50}$$



(a) "Explicit Solution" (b) MATLAB's pdepe

Fig. 3.7. Problems with state of the art approaches to solve the PDE for different initial states $X(0)$. (a) An "explicit" solution was proposed in [Xu+16, Eq. (18)], but provides only valid approximations for some states. For low initial states, the CDF is not even monotone. (b) Simulation results are approximated by a general purpose PDE solver [Xu+13], but suffer from severe oscillations and CDF values exceeding one.

Integrating over the entire cell provides the cell metric

$$P_i^{\mathrm{st}}(q_{\mathrm{a}}) = \int_{\mathcal{L}_i} P_{\boldsymbol{u},i}^{\mathrm{st}}(q_{\mathrm{a}}) f_{\boldsymbol{u},i}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}. \tag{3.51}$$

3.2.4 A Numerical Method to Solve the Involved PDEs

Whereas the systems (3.26) and (3.39) for $\boldsymbol{V}_{\boldsymbol{u},i}$ and $\boldsymbol{W}_{\boldsymbol{u},i}$ constitute linear ODE systems, with well-known explicit solutions, it is cumbersome to solve the PDE system (3.13) of coupled transport equations for $\boldsymbol{U}_{\boldsymbol{u},i}$. One-dimensional transport equations are relatively simple to be handled analytically. Under certain conditions, the principle can be transferred to multidimensional systems as well. Both approaches are described in Appendix A.4. However, the way the initial and boundary conditions (3.18) are formulated hinders simple analytical handling.

In [Xu+13; Xu+16], where a simpler situation, resulting in a system with only K_i equations and less coupling terms, was investigated, an "explicit solution" and the numerical approximation with MATLAB's general purpose numerical PDE solver `pdepe` were proposed. However, the proposed "explicit solution" was based on a mathematical error, what is shown in more detail in Appendix A.4.3. Thus, the proposed formula can only provide an approximation in the best

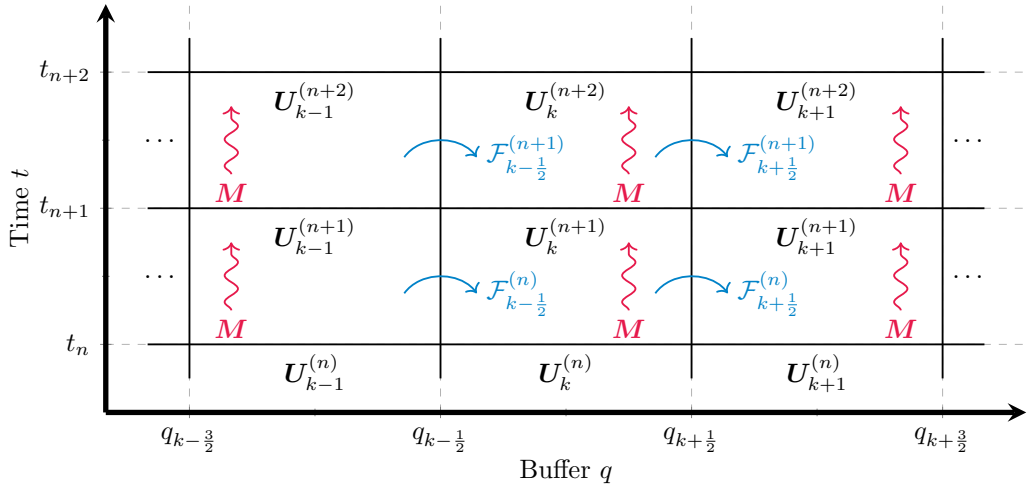


Fig. 3.8. Illustration of the implemented FVMs. The domain is discretized in time and space (buffer), such that the PDE solution is approximated for "finite volumes" with values $\boldsymbol{U}_k^{(n)} \cong \boldsymbol{U}(t_n, q_k)$. Whereas Eq. (3.52) approximates and applies the flux $\mathcal{F}$ between the volumes (blue), Eq. (3.53) handles the ODE part due to the transitions specified by $\boldsymbol{M}$ (red). The third dimension, i. e., the components of $\boldsymbol{U}$, is not depicted here.

case. In contrast, `pdepe` was designed to handle a wide range of different PDE types and so it is not specialized enough to treat transport equations with discontinuous boundary conditions. As a result, the `pdepe` approximation suffers from oscillations. The existing approaches are illustrated in Fig. 3.7. In Fig. 3.7a, it can be observed that the "explicit solution" is not accurate and creates non-monotone CDFs for low states. On the other hand, the result from `pdepe` shows severe oscillations for the curves of low states and results in CDFs exceeding one for the high states (cf. Fig. 3.7b).

In this thesis, the PDE system (3.13) is treated by a more appropriate numerical solver, namely a fractional step FVM [LeV07]. The principle of FVMs is sketched in Fig. 3.8. Its main idea is to discretize the geometry into small cells or volumes. The PDE describes a conservation law, which is then applied on the boundaries of the single volumes for each time step. The method, which is usually applied for fluid models in physics, is also appropriate for the situation at hand, due to the analogy to conservation laws described in footnote 4 on page 44. It is called fractional step method, because the homogeneous and inhomogeneous part of the PDE are handled separately in each time step. For the homogeneous part, the so-called *superbee* scheme [LeV07] is applied, which is a slope limiter that offers a trade-off between high order accuracy in smooth regions and avoidance of oscillations in the vicinity of discontinuities. The inhomogeneous coupling term on the RHS is then handled by an implicit Euler scheme.

The resulting two steps can be written as the following scheme

$$\tilde{\boldsymbol{U}}_k^{(n+1)} = \boldsymbol{U}_k^{(n)} - \boldsymbol{A}\frac{\Delta t}{\Delta q}\left[\left(\boldsymbol{U}_k^{(n)} - \boldsymbol{U}_{k-1}^{(n)}\right) - \frac{1}{2}\left(\Delta q \boldsymbol{I} - \boldsymbol{A}\Delta t\right)\left(\sigma_k^n - \sigma_{k-1}^n\right)\right], \quad (3.52)$$

$$\boldsymbol{U}_k^{(n+1)} = \left(\boldsymbol{I} - \Delta t \boldsymbol{M}\right)^{-1} \tilde{\boldsymbol{U}}_k^{(n+1)}. \quad (3.53)$$

Here, $\boldsymbol{U}_k^{(n+1)}$ refers to the discretized solution at the k^{th} volume in the $(n+1)^{\text{th}}$ time step, whereas the tilde indicates the intermediate result of the first step. All other indices are omitted for the sake of notation. The variable σ_k^n denotes the slope between adjacent cells, which is limited by the superbee scheme. Further details and explanations can be found in the Appendix A.5.

Furthermore, two approximations are studied for performance gains, as described in the subsequent sections.

No Dynamics

A simple model with an explicit solution can be derived, if all possible transitions are removed, such that interference and load conditions are static during the prefetching and playback and can only differ upon arrival. Thus, by removing all linear coupling terms in Eq. (3.13), (3.25), and (3.36) from the transition matrices $\boldsymbol{M}_{\boldsymbol{u},i}$, and $\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}$, respectively, decoupled PDEs and ODEs are obtained for each state $(\boldsymbol{y}, z_i)$:

$$\frac{\partial}{\partial t} U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q) + v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} \frac{\partial}{\partial q} U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q) = 0, \quad (3.54)$$

$$\frac{\mathrm{d}}{\mathrm{d}q} V_{(\boldsymbol{y},z_i)}^{(\boldsymbol{\xi},\eta_i)}(q; q_\mathrm{a}) = 0, \quad (3.55)$$

$$\frac{\mathrm{d}}{\mathrm{d}q} W^{(\boldsymbol{y},z_i)}(q) = 0. \quad (3.56)$$

This way, the PDE becomes a one-dimensional linear transport equation with a solution given by

$$U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}(t,q) = \begin{cases} 1, & q - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} t \leq 0, \\ 0, & q - v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} t > 0, \end{cases} \quad (3.57)$$

whereas $V_{(\boldsymbol{y},z_i)}^{(\boldsymbol{\xi},\eta_i)}$ and $W^{(\boldsymbol{y},z_i)}$ become constant functions that depend only on their initial conditions. Thus, the state transition probabilities $\boldsymbol{V}_{\boldsymbol{u},i}$ will stay a unit matrix. The starvation probabilities $W^{(\boldsymbol{y},z_i)}$ will be zero for any non-negative rate $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$. For negative rates, the starvation probability will be the probability

that the video length exceeds the length the buffer needs to run empty with the constant effective rate $w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$.

$$W^{(\boldsymbol{y},z_i)}(q) = \begin{cases} 0, & w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} \geq 0, \\ \exp\left(-\bar{T}_{\text{video}}^{-1} \cdot \dfrac{q}{|w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}|}\right), & w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)} < 0. \end{cases} \tag{3.58}$$

Semi-Dynamic

Another approximating model can be obtained, by removing all transitions between different interference scenarios, i. e., to model the interference as quasi-stationary during prefetching and playback, and thus, to consider the flow dynamics only, and so leading to a similar system to the one in [KF15]. For the sparsity pattern in Fig. 3.5 this means to remove all coupling terms indicated by blue markers.

This results in L decoupled PDE systems of size K for $\boldsymbol{U}_{\boldsymbol{u},i}$ and respective ODE systems for $\boldsymbol{V}_{\boldsymbol{u},i}$ and $\boldsymbol{W}_{\boldsymbol{u},i}$ for each interference state. The PDE systems can then be solved with the numerical FVM approach described at the beginning of this section, whereas the ODE systems have explicit solutions given in Eqns. (3.30) and (3.44), respectively. Thanks to the decoupling, the L systems can be solved in parallel, if there is sufficient computation capacity available.

Fully Dynamic

The fully dynamic case describes the models as originally derived with all coupling terms. The solution can be obtained with the same methods as in the semi-dynamic case, but here no advantage can be taken from parallelization of independent systems in a simple way.

3.2.5 Results and Numerical Validation

The presented model allows to conduct a series of studies to assess the impact of different system parameters on video streaming. For instance, the influence of admission control, different traffic demands in neighboring cells, or the video bitrate was investigated in [Sch+20b]. Also a user-centric QoE metric was proposed and presented in [SKF17]. However, this thesis only focuses on the model validation with the help of system-level simulations and on the trade-off that can be observed between both video performance metrics.

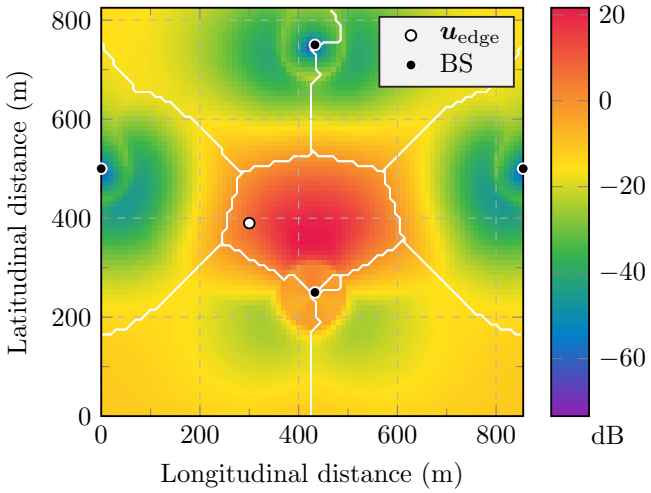
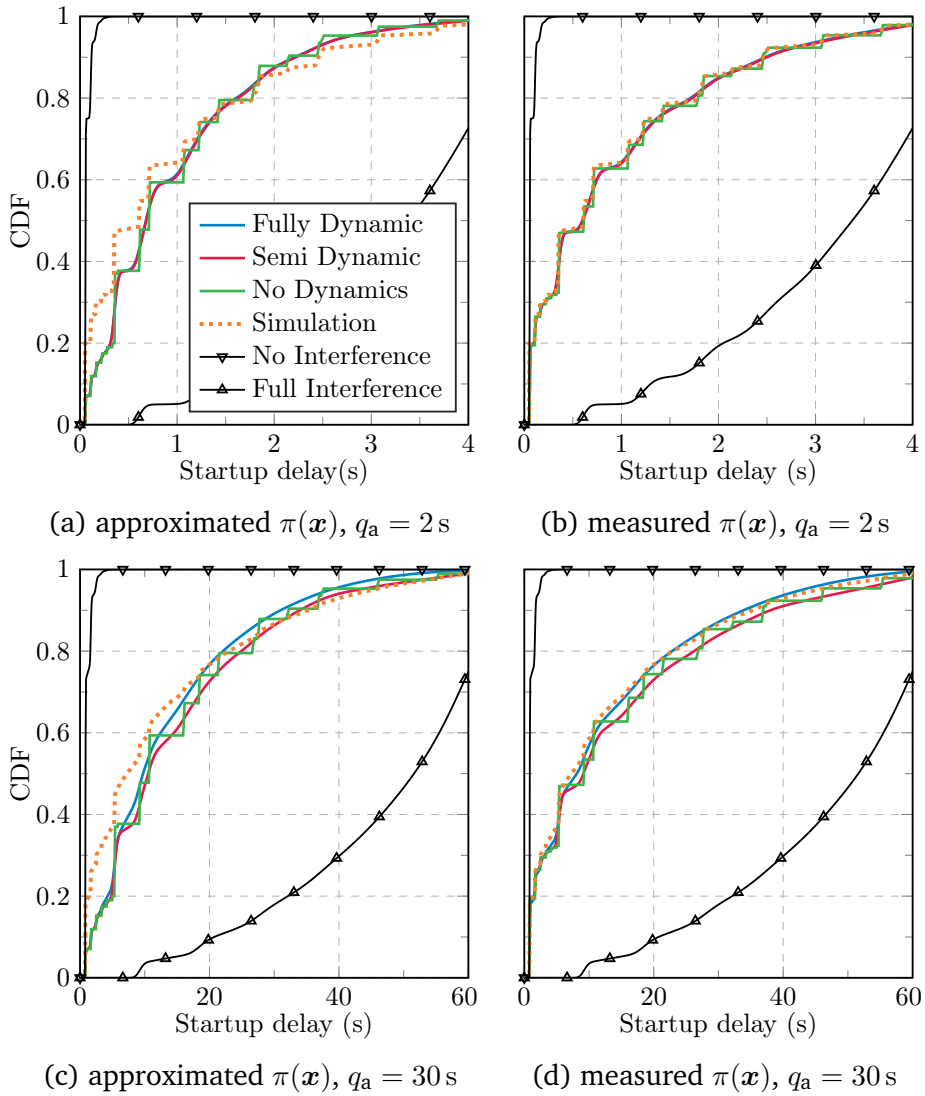


Fig. 3.9. Illustration of the evaluated scenario. The figure shows the SINR map of the considered cell according to the assumed path loss model in Eq. (3.4) and with full interference. A white and four black circles indicate the considered edge location $\boldsymbol{u}_{\text{edge}}$ and the BS sites, respectively. The white lines depict the cell boundaries according to the strongest received power.

For this purpose, a cellular network scenario consisting of seven sectorized hexagonal macro cells with an inter-site distance of $500\,\text{m}$ is considered. As in [Kle+16], the network parameters, e. g., path loss, shadowing, antenna patterns, are aligned with the 3GPP recommendations in [3GP10]. The video traffic is characterized by an exponentially distributed video length with a mean of $\bar{T}_{\text{video}} = 240\,\text{s}$ and a video bitrate of $R_{\text{CBR}} = 2\,\text{Mbps}$. The traffic in the considered cell and in the neighboring cells are assumed to be $T_{\text{center}} = T_{\text{neighbor}} = 18\,\text{Mbps}$, if not stated otherwise. The admission control is set to $K_i = 7$ for all $i \in \mathcal{N}$. For brevity, all simulation parameters are summarized in Table A.1 in Appendix A.1.

A TTI-based system-level simulator was used to validate the model results. To generate the statistics of cell-related metrics, approximately $N_{\text{F}} = 7.5 \times 10^6$ video flows were generated within the considered cell, which corresponds to a simulated real time of roughly $55\,000$ hours. For the location-dependent metrics, it is ensured that at least $10\,000$ flows were generated close to a pre-defined position $\boldsymbol{u}_{\text{edge}}$ near the cell edge (cf. Fig. 3.9). Mobility is not considered under the assumption that most of the traffic is generated by static users. The model results are based on an FVM discretization with $N_{\text{G}} = 16\,000$ grid cells.



(a) approximated $\pi(\boldsymbol{x})$, $q_\mathrm{a} = 2\,\mathrm{s}$ (b) measured $\pi(\boldsymbol{x})$, $q_\mathrm{a} = 2\,\mathrm{s}$

(c) approximated $\pi(\boldsymbol{x})$, $q_\mathrm{a} = 30\,\mathrm{s}$ (d) measured $\pi(\boldsymbol{x})$, $q_\mathrm{a} = 30\,\mathrm{s}$

Fig. 3.10. Model validation for the startup delay distribution at a cell edge location, conducted for two different buffer threshold values of $q_\mathrm{a} = 2\,\mathrm{s}$ in (a) and (b), as well as $q_\mathrm{a} = 30\,\mathrm{s}$ in (c) and (d), respectively. The results in (a) and (c) were created with modeled state probabilities, whereas (b) and (d) rely on measured probabilities by simulation. All plots contain the bounding regimes of no and full interference as well.

Startup Delay Results

The created CDFs for the startup delay distribution at the location $\boldsymbol{u}_\mathrm{edge}$ from the different models and simulation are depicted in Fig. 3.10. The plots also contain curves with the bounding no and full interference regimes, which can be obtained by fixing the interference scenario to $\boldsymbol{y} = \boldsymbol{e}_i$ and $\boldsymbol{y} = \mathbf{1}$ resulting in worst and best receiving conditions, respectively. In subplots 3.10a–b the buffer was set to a relatively short length of $q_\mathrm{a} = 2\,\mathrm{s}$. Here, all approaches lead to similar results, which can be explained by the fact that only few state

transitions may happen in such a short buffering time and, thus, almost static conditions. This is also the reason for the CDFs forming typical step functions, where each step represents one of the states $(\boldsymbol{y}, z_i)$ with the respective receiving conditions. For the model without any dynamics, those steps are always sharply defined, because there are no possible transitions which would smoothen the CDF. In subfigure 3.10a, a small gap between the simulated and all of the model curves can be observed. The cause of this error was identified as the inaccuracies resulting from the state aggregation method to obtain the state probabilities, which act as weighting factors for the different steps of the CDF. This can be seen by inspecting subfigure 3.10b, where the modeled state probabilities were replaced by the "true" probabilities obtained from the simulation.

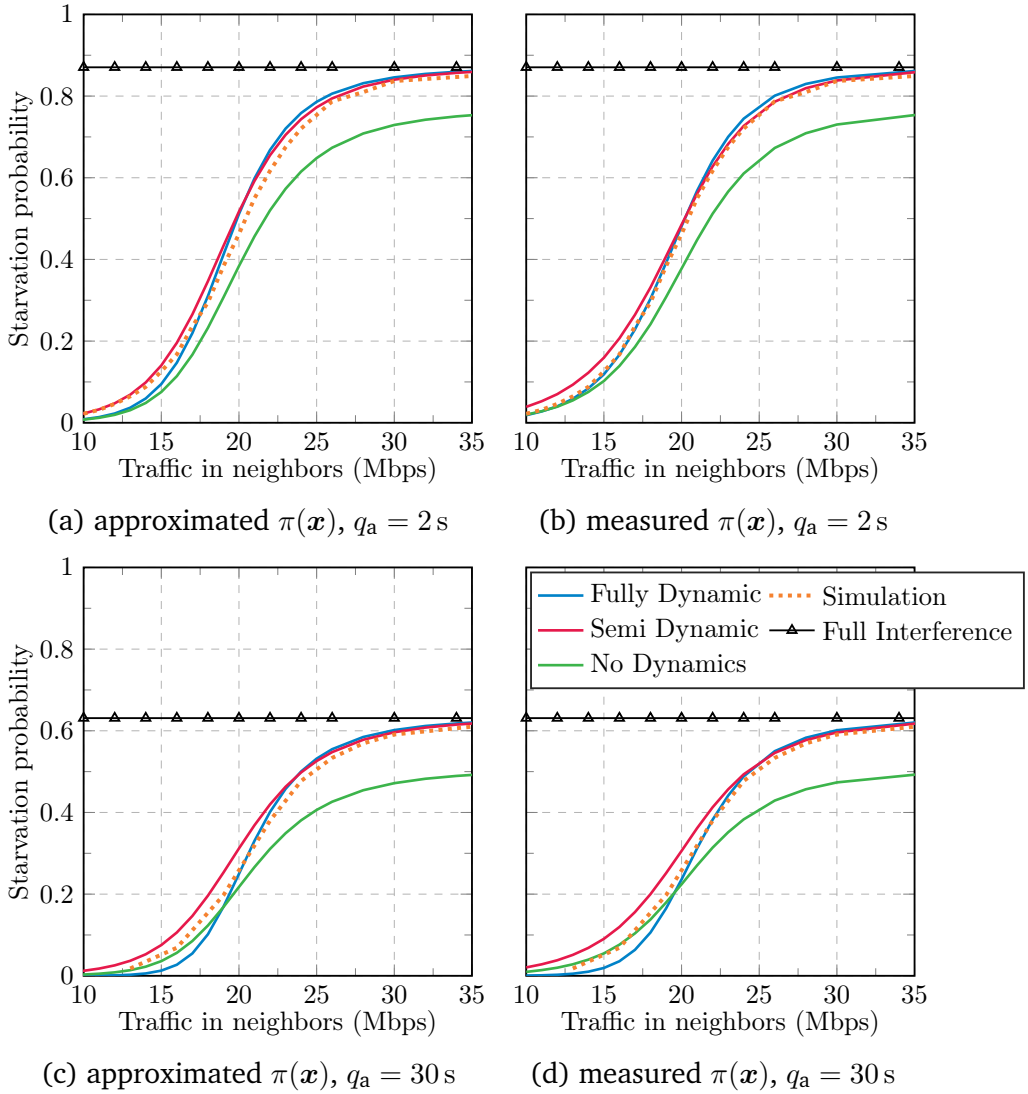
Subfigures 3.10c–d show analog results, but for a higher buffering threshold of $q_\mathrm{a} = 30\,\mathrm{s}$. Clearly, the longer threshold leads to potentially more transitions and, thus, to smoother CDFs curves. Consequently, the fully dynamic model leads to the most accurate approximation of the simulated curve. In contrast, the model without any dynamics still forms sharp steps due to its definition.

Starvation Probability Results

Fig. 3.11 shows the results for the starvation probability. Again, simulation results are compared to the different models and the bounding regime of full interference[6] for two different buffer thresholds $q_\mathrm{a} \in \{2, 30\}\,\mathrm{s}$ and with modeled and simulated probabilities $\pi(\boldsymbol{x})$. The starvation probability is shown for different traffic demands in the neighboring cells, which directly translates to different arrival rates in the neighbor cells, because other video parameters are fixed.

It can be observed that the impact of the state probabilities is not as severe as in the case of the startup delay. In addition, a clear inaccuracy of the approach without any dynamics can be observed. Both can be explained as follows. The starvation probability is a metric of the entire time span of the video playback. Thus, it depends more on the process dynamics than on the initial states, which cannot be captured by the model without dynamics. There is even a self-reinforcing effect, called *self-interference*. Whenever a flow arrives to the system, its presence degrades the performance of flows at the same but also at neighboring BSs. This degrades the performance of all other flows, such that they stay longer in the system and, thus, in turn worsen the experienced

[6] The upper perfomance bound of no interference would lead to a starvation probability of zero and is therefore not explicitly drawn.



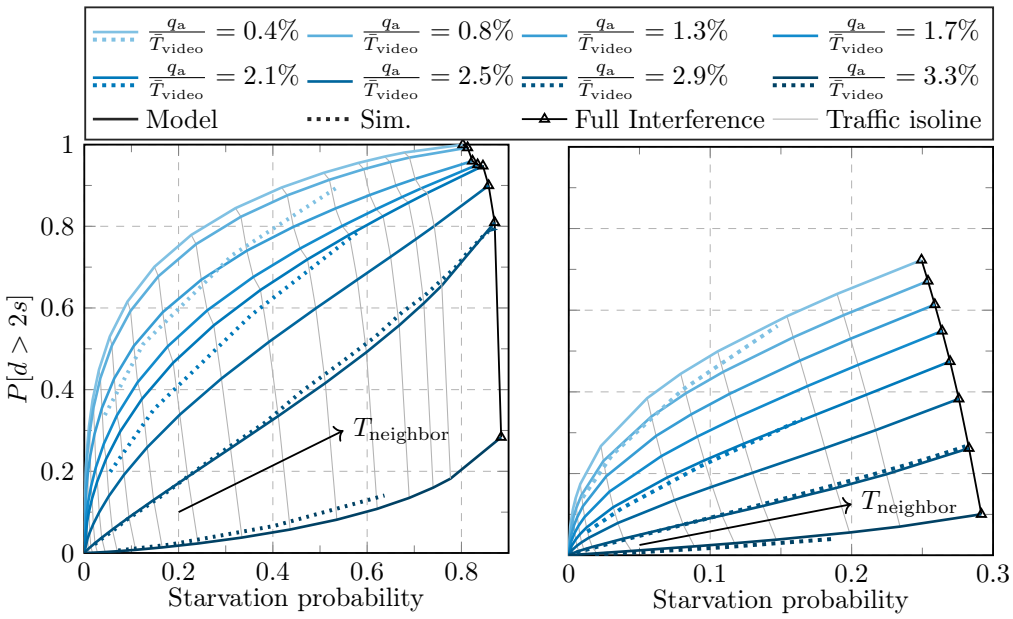
(a) approximated $\pi(\boldsymbol{x})$, $q_\mathrm{a} = 2\,\mathrm{s}$ (b) measured $\pi(\boldsymbol{x})$, $q_\mathrm{a} = 2\,\mathrm{s}$

(c) approximated $\pi(\boldsymbol{x})$, $q_\mathrm{a} = 30\,\mathrm{s}$ (d) measured $\pi(\boldsymbol{x})$, $q_\mathrm{a} = 30\,\mathrm{s}$

Fig. 3.11. Model validation for the starvation probability at different neighbor traffic demands at a cell edge location. It was conducted for two different buffer threshold values of $q_\mathrm{a} = 2\,\mathrm{s}$ in (a) and (b), as well as $q_\mathrm{a} = 30\,\mathrm{s}$ in (c) and (d), respectively. The results in (a) and (c) were created with modeled state probabilities, whereas (b) and (d) rely on measured values by simulation. All plots contain the bounding regimes of full interference as well.

interference. Because the model without dynamic cannot capture such effects, it overestimates the performance.

In contrast, the semi and fully dynamic model, provide good approximations of simulated results. It can also be observed that for high traffic demand in neighboring cells, the performance converges to the full interference bound, because for high traffic, all BSs are likely to be active.



(a) edge (b) cell

Fig. 3.12. Tradeoffs between startup delay and starvation probability at (a) a cell edge location and (b) over the entire cell. The figure shows data points for different values of the buffer threshold and increasing data traffic in neighboring cells. The gray isolines connect model results obtained with the same neighbor traffic demand.

Tradeoff between Startup Delay and Starvation Probability

As it could be seen above, the startup buffer threshold q_a has an influence of the startup delay CDFs, which are shifted towards longer delays for higher values of q_a, and on the starvation probability, which is decreased for higher values. Thus, when choosing a value for q_a, there is a tradeoff between both KPIs, which is illustrated in Fig. 3.12. Different data points represent different values $q_a \in \{1, \ldots, 8\}\,\mathrm{s}$, which are normalized by the mean video length to $\frac{q_a}{\bar{T}_{video}} \in \{0.4, \ldots, 3.3\}\,\%$, on the one hand. On the other hand, they refer to different traffic in neighbor cells, which is indicated by gray isolines. On the x-axis they mark the resulting starvation probability whereas on the y-axis, the probability of exceeding a certain value of buffering time, namely $2\,\mathrm{s}$, is depicted. The value was chosen according to the studies in [KS12]. Furthermore, the full interference bound is drawn on the right, to which all curves converge for increasing traffic.

It turns out, that even small deviations of q_a have a strong impact on the startup delay, but only minor effect on the starvation probability, as indicated by the almost vertical traffic isolines. As it can be seen in Fig. 3.11, large buffers have to

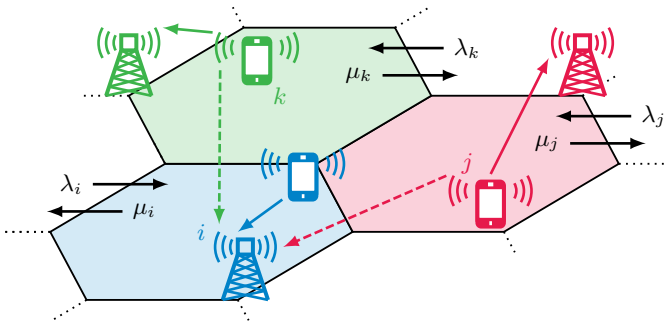


Fig. 3.13. The dynamics in a cellular network for the UL. The scenario is very simular to the DL depicted in Fig. 3.1. However, interference (dashed arrows) now stems from other UEs that can have various (combinations of) positions. In contrast, in the DL the positions of the interfering BSs were fixed.

be filled for a significant decrement of the starvation probability. Therefore, other means to reduce the starvation probability (e. g., reducing the video bitrate), have to be considered. Further details on the study are given in [Sch+20b].

3.3 Extension to the Uplink

Flow-level models have found wide application for analyzing the DL of cellular systems (e. g., [BBP04; Bon04b; Bon05; KFF14; KGF14; Kle+16; KF15]). However, to the best of the author's knowledge only little effort was spent on the UL. For instance, the authors in [EE10] propose a UL flow-level model, but consider interference as a scaled version of the received power only. In this section, an approach is introduced to fill the gap. It turns out that in the UL the interference modeling becomes much more complex than in the DL, which may be the reason that there is only little research conducted in this area until now.

For the study, a similar scenario as for the DL is under investigation as depicted in Fig. 3.13. The main difference compared with the DL is that interference now stems from other UEs with flexible positions that have to be taken into account and not from BSs with fixed locations. For this work it is assumed that there is no mobility and users are multiplexed by time division, such that there is only one user transmitting in each time slot per cell at maximum. Let the matrix $\boldsymbol{v} = (\boldsymbol{u}_i) \in \mathscr{L} := \bigtimes_{i \in \mathcal{N}} \mathcal{L}_i \subseteq \mathbb{R}^{N \times 2}$ collect the potential positions of one active user in each cell[7]. Further, let $p_{ij}^{\mathrm{rx}}(u)$ denote the power that BS i receives at its antenna from the active user in cell j.

[7] The matrix $\boldsymbol{v}$ always collects one position for each cell, no matter whether there is an active user or not. If the cell j is inactive, than this position will be ignored, due to the factor $y_j = 0$.

Assumption 3.3 (perfect power control)**.** *Each BS applies perfect power control to the transmit power of each associated user, such that it receives* $p_{ii}^{\mathrm{rx}}(\boldsymbol{u}) = \bar{p}_i^{\mathrm{rx}}$ *from all users in the own cell.*

Based on these assumptions and definitions, the SINR γ_i received at BS i can be stated as follows

$$\gamma_i(\boldsymbol{\upsilon}, \boldsymbol{y}) = \frac{p_{ii}^{\mathrm{rx}}(\boldsymbol{\upsilon}_i)}{\sum_{j \in \mathcal{N}_{-i}} y_j p_{ij}^{\mathrm{rx}}(\boldsymbol{\upsilon}_j) + N_0} = \frac{\bar{p}_i^{\mathrm{rx}}}{\sum_{j \in \mathcal{N}_{-i}} y_j p_{ij}^{\mathrm{rx}}(\boldsymbol{\upsilon}_j) + N_0} \tag{3.59}$$

and, thus, the maximum achievable rate can be expressed as

$$c_i(\boldsymbol{\upsilon}, \boldsymbol{y}) = B_{\mathrm{BW}} \log_2 \left(1 + \gamma_i(\boldsymbol{\upsilon}, \boldsymbol{y})\right) \tag{3.60}$$

based on Shannon's law. Similar to the DL model (cf. Eg. (3.7)), this formula could be modified with bandwidth and SINR efficiencies and cut at a maximum to be tailored to a realistic wireless system.

Both parameters, $\boldsymbol{\upsilon}$ and $\boldsymbol{y}$ follow random processes and are therefore realizations of the RVs $\boldsymbol{\Upsilon}$ and $\boldsymbol{Y}$, respectively. As with the DL case, a BS location independent service rate μ_i shall be derived. Thus, the goal is now to obtain the average data rate or capacity independent of $\boldsymbol{\upsilon}$, i. e., to find $\mathbb{E}\left[c_i(\boldsymbol{\Upsilon}, \boldsymbol{Y})\right]$.

The user locations in cell i follow a given distribution with PDF $f_{\boldsymbol{u},i}$. It is reasonable to assume user locations in different cells mutually independent and so to consider the joint distribution as the product of the individual PDFs

$$f_{\boldsymbol{\upsilon}}(\boldsymbol{\upsilon}) := \prod_{i \in \mathcal{N}} f_{\boldsymbol{u},i}(\boldsymbol{\upsilon}_i) \tag{3.61}$$

The following notations are introduced for the density functions of the marginal distributions of users in

$$\text{cell } i\text{'s neighbors} \qquad f_{\boldsymbol{\upsilon},-i}(\boldsymbol{\upsilon}) := \prod_{j \in \mathcal{N}_{-i}} f_{\boldsymbol{u},j}(\boldsymbol{\upsilon}_j), \text{ and in} \tag{3.62}$$

$$\text{cell } i\text{'s active neighbors} \qquad f_{\boldsymbol{\upsilon},i,\boldsymbol{y}}(\boldsymbol{\upsilon}) := \prod_{j \in \mathcal{N}_{i,1}(\boldsymbol{y})} f_{\boldsymbol{u},j}(\boldsymbol{\upsilon}_j), \tag{3.63}$$

respectively. In addition, let the terms

$$f_{\boldsymbol{v}}(\boldsymbol{v},\boldsymbol{y}) := \mathbb{P}\left[\boldsymbol{Y}=\boldsymbol{y}\right]\prod_{i\in\mathcal{N}} f_{\boldsymbol{u},i}(\boldsymbol{v}_i), \tag{3.64}$$

$$f_{\boldsymbol{v},-i}(\boldsymbol{v},\boldsymbol{y}) := \mathbb{P}\left[\boldsymbol{Y}=\boldsymbol{y}\right]\prod_{j\in\mathcal{N}_{-i}} f_{\boldsymbol{u},j}(\boldsymbol{v}_j), \text{and} \tag{3.65}$$

$$f_{\boldsymbol{v},i,\boldsymbol{y}}(\boldsymbol{v},\boldsymbol{y}) := \mathbb{P}\left[\boldsymbol{Y}=\boldsymbol{y}\right]\prod_{j\in\mathcal{N}_{i,1}(\boldsymbol{y})} f_{\boldsymbol{u},j}(\boldsymbol{v}_j) \tag{3.66}$$

denote the joint distribution of $\boldsymbol{\Upsilon}$ and $\boldsymbol{Y}$, the joint distribution of the users in cell i's neighbors and $\boldsymbol{Y}$, and the joint distribution of the users in cell i's active neighbors and $\boldsymbol{Y}$, respectively. In this regard, the cartesian product of the active neighbor cells is denoted by

$$\mathscr{L}_{i,\boldsymbol{y}} := \bigtimes_{j\in\mathcal{N}_{i,1}} \mathcal{L}_j. \tag{3.67}$$

Inspecting the achievable rate in Eq. (3.60), which depends on user locations and the interference scenario, it can be observed that the user's performances are mutually coupled. Especially users close to the cell edge try to overcome large path loss with high transmit power and thereby degrade the experienced SINR in neighboring cells. This degrades neighbor's performance and, in turn, increases the probability of experiencing interference in the considered cell (cf. self-interference, p. 57). With this and ξ denoting the random flow size with mean Ω, the average service time becomes

$$\frac{1}{\mu_i} = \mathbb{E}\left[\frac{\xi}{c_i(\boldsymbol{\Upsilon},\boldsymbol{Y})}\right] \approx \Omega\cdot\mathbb{E}_{\boldsymbol{\Upsilon}_i}\left[\frac{1}{\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}\left[c_i(\boldsymbol{\Upsilon},\boldsymbol{Y})\mid\boldsymbol{\Upsilon}_i=\boldsymbol{u},Y_i=1\right]}\right], \tag{3.68}$$

such that the average service rate can be approximated as

$$\mu_i \approx \frac{1}{\Omega}\left[\mathbb{E}_{\boldsymbol{\Upsilon}_i}\left[\frac{1}{\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}\left[c_i(\boldsymbol{\Upsilon},\boldsymbol{Y})\mid\boldsymbol{\Upsilon}_i=\boldsymbol{u},Y_i=1\right]}\right]\right]^{-1} \tag{3.69}$$

$$\approx \frac{1}{\Omega}\left[\int_{\mathcal{L}_i}\frac{f_{\boldsymbol{u},i}(\boldsymbol{u})}{\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}\left[c_i(\boldsymbol{\Upsilon},\boldsymbol{Y})\mid\boldsymbol{\Upsilon}_i=\boldsymbol{u},Y_i=1\right]}\mathrm{d}\boldsymbol{u}\right]^{-1}. \tag{3.70}$$

Here, $\mathbb{E}_{\boldsymbol{\Upsilon}_i}[\cdot]$ and $\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}[\cdot]$ denote the expectation with respect to the random location in the considered cell i and with respect to the locations in all other cells, respectively.

The expectation value in the denominator of the integrand constitutes a fluid regime assumption (cf. [Kle+16]). This essentially means that the considered

flow observes all possible interference constellations $\boldsymbol{v} \in \mathscr{L}_i(\boldsymbol{u}) := \{\boldsymbol{v} \in \mathscr{L} | \boldsymbol{v}_i = \boldsymbol{u}\}$, $\boldsymbol{y} \in \mathcal{Y}_{-i} := \{\boldsymbol{y} \in \mathcal{Y} | y_i = 1\}$ in the neighboring cells infinitely often and infinitely fast. The fact that a fluid regime approximation is applied leads to an overestimation of the actual performance and is expected to result in a non-negligible error.

Together with Poissonian arrivals with rates λ_i and the mean service rates μ_i, an $\mathrm{M}/\mathrm{M}/1/\infty$ PS model is assumed. Thus, the probability that BS i is active becomes

$$\mathbb{P}[Y_i = 1] = \min\{\rho_i, 1\}. \tag{3.71}$$

The system is also assumed to be stable, i. e., $\rho_i < 1$, as discussed earlier. The BS activity depends on the distribution of $\boldsymbol{\Upsilon}$ and $\boldsymbol{Y}$, which leads to an interference-coupled system. However, for the joint distribution of the interference, the following approximation is made

$$\begin{aligned}
\mathbb{P}[\boldsymbol{Y} = \boldsymbol{y}] &\approx \prod_{i \in \mathcal{N}} \mathbb{P}[Y_i = y_i] && (3.72)\\
&= \prod_{j \in \mathcal{N}_{i,1}(\boldsymbol{y})} \mathbb{P}[Y_i = 1] \prod_{j \in \mathcal{N}_{i,0}(\boldsymbol{y})} \mathbb{P}[Y_i = 0] && (3.73)\\
&= \prod_{j \in \mathcal{N}_{i,1}(\boldsymbol{y})} \rho_j \cdot \prod_{j \in \mathcal{N}_{i,0}(\boldsymbol{y})} (1 - \rho_j). && (3.74)
\end{aligned}$$

In this approximation, the BS activities are assumed to be mutually independent (Eq. (3.72)). This is actually only true for the user arrivals to each BS, but does not incorporate the effects of self-interference (cf. p. 57), which will be only incorporated via the derived load factors ρ_i. It is assumed that the joint probability of BS activities is dominated by the independent user arrivals. The product was split into the components referring to active and inactive cells (Eq. (3.73)) such that the BS loads can be inserted accordingly (Eq. (3.74)).

With this, the expected fluid achievable rate can be derived, which is conditioned on the fact that the BS of the considered user in cell i is active

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}[c_i(\boldsymbol{\Upsilon}, \boldsymbol{Y}) \mid \boldsymbol{\Upsilon}_i = \boldsymbol{u}, Y_i = 1] &= \sum_{\boldsymbol{y} \in \mathcal{Y}_i} \int_{\mathscr{L}_i(\boldsymbol{u})} c_i(\boldsymbol{v}, \boldsymbol{y}) f_{\boldsymbol{v},-i}(\boldsymbol{v}, \boldsymbol{y}) \mathrm{d}\boldsymbol{v} && (3.75a)\\
&= \sum_{\boldsymbol{y} \in \mathcal{Y}_i} \mathbb{P}[\boldsymbol{Y} = \boldsymbol{y}] \int_{\mathscr{L}_i(\boldsymbol{u})} c_i(\boldsymbol{v}, \boldsymbol{y}) f_{\boldsymbol{v},-i}(\boldsymbol{v}) \mathrm{d}\boldsymbol{v}. && (3.75b)\\
&= \sum_{\boldsymbol{y} \in \mathcal{Y}_i} \mathbb{P}[\boldsymbol{Y} = \boldsymbol{y}] \int_{\mathscr{L}_{i,\boldsymbol{y}}} c_i(\boldsymbol{v}, \boldsymbol{y}) f_{\boldsymbol{v},i,\boldsymbol{y}}(\boldsymbol{v}) \mathrm{d}\boldsymbol{v}. && (3.75c)
\end{aligned}$$

The steps can be explained as follows. In Eq. (3.75a) the definition of the expectation value with respect to $\boldsymbol{\Upsilon}_{-i}$ is written down by summing over the interference scenarios and user locations with respect to the corresponding joint PDF. The PDF is then split into its components in Eq. (3.75b). Finally, the integration domain is restricted to the active components in Eq. (3.75c).

In the system (3.75), the integrals are only meant to be taken over the components that are not fixed (e. g., the location of the active user in the considered cell $\boldsymbol{u}_i$). In particular, the integrals in the previous equations are at most $(N-1)$-dimensional. For each inactive cell, i. e., each j with $y_j = 0$, the integrand does not depend on $\boldsymbol{v}_j$, and so the dimensionality decreases by one. For the sake of simplicity, no further notation is introduced to take this into account.

Due to the power-control (Assumption 3.3), the RHS in Eq. (3.75c) is independent from $\boldsymbol{u}$ and can be inserted into Eq. (3.70) to obtain the mean service rate

$$\mu_i \approx \frac{1}{\Omega}\left[\int_{\mathcal{L}_i} \frac{f_{\boldsymbol{u},i}(\boldsymbol{u})}{\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}\left[c_i(\boldsymbol{\Upsilon},\boldsymbol{Y}) \,|\, \boldsymbol{\Upsilon}_i = \boldsymbol{u}, Y_i = 1\right]} \mathrm{d}\boldsymbol{u}\right]^{-1} \tag{3.76}$$

$$= \frac{1}{\Omega}\left[\int_{\mathcal{L}_i} \frac{f_{\boldsymbol{u},i}(\boldsymbol{u})}{\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}\left[c_i(\boldsymbol{\Upsilon},\boldsymbol{Y}) \,|\, Y_i = 1\right]} \mathrm{d}\boldsymbol{u}\right]^{-1} \tag{3.77}$$

$$= \frac{1}{\Omega}\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}\left[c_i(\boldsymbol{\Upsilon},\boldsymbol{Y}) \,|\, Y_i = 1\right]\left[\int_{\mathcal{L}_i} f_{\boldsymbol{u},i}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}\right]^{-1} \tag{3.78}$$

$$= \frac{1}{\Omega}\mathbb{E}_{\boldsymbol{\Upsilon}_{-i}}\left[c_i(\boldsymbol{\Upsilon},\boldsymbol{Y}) \,|\, Y_i = 1\right] \tag{3.79}$$

$$= \frac{1}{\Omega}\sum_{\boldsymbol{y}\in\mathcal{Y}_i} \mathbb{P}\left[\boldsymbol{Y} = \boldsymbol{y}\right] \int_{\mathscr{L}_{i,\boldsymbol{y}}} c_i(\boldsymbol{v},\boldsymbol{y}) f_{\boldsymbol{v},i,\boldsymbol{y}}(\boldsymbol{v})\mathrm{d}\boldsymbol{v}. \tag{3.80}$$

Thanks to the independence from $\boldsymbol{u}$, the condition on $\boldsymbol{u}$ can be removed in Eq. (3.77) to obtain Eq. (3.77), allowing to extract the expectation value from the integral (Eq. (3.78)). The integral over the PDF is one (Eq. (3.79)). Finally, inserting Eq. (3.75c) provides the result in Eq. (3.80) considering the independence from $\boldsymbol{u}$.

Inserting the interference state probabilities from Eq. (3.74) leads to one nonlinear equation for each $i \in \mathcal{N}$

$$\mu_i \approx \frac{1}{\Omega}\sum_{\boldsymbol{y}\in\mathcal{Y}_i} \prod_{j\in\mathcal{N}_{i,1}} \rho_j \prod_{j\in\mathcal{N}_{i,0}} (1-\rho_j) \int_{\mathscr{L}_{i,\boldsymbol{y}}} c_i(\boldsymbol{v},\boldsymbol{y}) f_{\boldsymbol{v},i,\boldsymbol{y}}(\boldsymbol{v})\mathrm{d}\boldsymbol{v}, \tag{3.81}$$

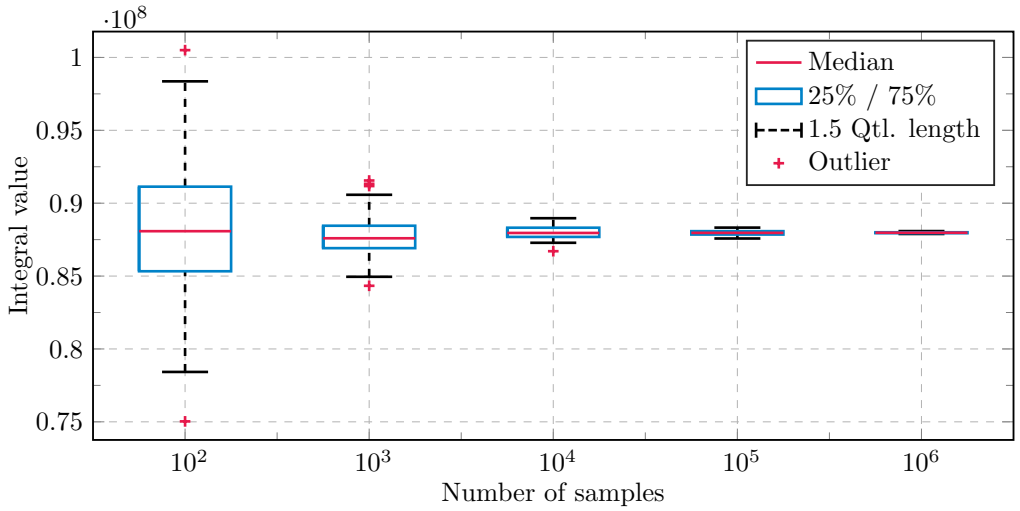


Fig. 3.14. Convergence of a Monte Carlo integration for evaluating the coefficients with multi-dimensional integrals in Eq. (3.82).

which can be reformulated by replacing $\mu_i = \frac{\lambda_i}{\rho_i}$ and inverting the entire equation to obtain

$$\rho_i \approx \lambda_i \Omega \left[\sum_{\boldsymbol{y} \in \mathcal{Y}_i} \prod_{j \in \mathcal{N}_{i,1}} \rho_j \prod_{j \in \mathcal{N}_{i,0}} (1 - \rho_j) \int_{\mathscr{L}_{i,\boldsymbol{y}}} c_i(\boldsymbol{v}, \boldsymbol{y}) f_{\boldsymbol{v},i,\boldsymbol{y}}(\boldsymbol{v}) \mathrm{d}\boldsymbol{v} \right]^{-1}. \tag{3.82}$$

Let $\bar{\rho}_i$ denote the solution of the non-linear system (3.82) with N equations and N unknown variables ρ_i. Under the assumption of dealing with $\mathrm{M}/\mathrm{M}/1/\infty$ PS queuing systems, the steady-state probability distribution becomes

$$\mathbb{P}\left[\boldsymbol{X}_i = x_i\right] \approx \bar{\rho}_i^{x_i} \left(1 - \bar{\rho}_i\right). \tag{3.83}$$

Furthermore, interesting network- and user-centric KPIs can be assessed with the formulas given in Table 3.2.

The system (3.82) is non-linear and contains addends with multi-dimensional integrals. Here, the number of addends grows exponentially with the number of cells N and the maximum number of the integral's dimensions grows linear with N. Therefore, evaluation is only feasible for scenarios with low complexity and, thus, the model is more of theoretical interest. For instance, Fig. 3.14 shows that the integrals can be approximated with a Monte Carlo integrator for a small scenario with $N = 3$ cells. However, further research has to put effort on finding reasonable model approximations to increase the practical relevance.

3.4 Modeling versus Simulation

One major motivation behind modeling approaches is to avoid exhaustive simulation or even experimental times. Therefore, this section aims to compare the developed models with simulation with respect to computational complexity as well as actually measured computation times for an exemplary scenario.

It should be noted that it is not a trivial task to compare both approaches, because both can be parameterized with an impact on the accuracy. On the one hand, accuracy of a simulation increases with its duration. On the other hand, the accuracy of numerical approximation with FVM depends on the granularity of the discretization. Thus, a fair comparison has to take both into account, accuracy and duration. However, for the scenario at hand, there exist no true values, which can be taken as a benchmark. Also, by its nature, simulation results converge fast in the region of lower percentiles, but struggle when it comes to the distribution's tail, because a tremendous number of runs is required to obtain reliable results for very high percentiles. In contrast, the model aims for the entire distribution. Lastly, the simulation has to be run for the entire scenario, whereas the model can be evaluated for individual locations only, if a single location-dependent metric is of interest. Thus, a comparison also depends on which particular metric is of interest.

The following complexity analysis focuses on the DL streaming model and in particular on the parts that are responsible for the major contributions to the computation time. The analysis does not include any potential parallelization gains. Its findings are summarized in Table 3.3.

Simulation

The complexity of the TTI-based simulation is dominated by the number of cells N and the number of TTIs that have to be simulated, which is the product of the simulated real time T_{real} and the inverse of the TTI length T_{TTI}. Furthermore, after the simulation run, the PDFs have to be calculated, e. g., by using a kernel-based density estimator. Its complexity depends on the number of generated flows $N_{\mathrm{F}} \cong T_{\mathrm{real}}\lambda_i$ and the number of discretized cells N_{G}. If only a certain location is of interest, the last part reduces by the number of discretized locations $N_{\boldsymbol{u}}$.

Tab. 3.3. Simulation versus model. A comparison of complexity $\mathscr{C}$ and computation time T_{cp}.

	Simulation		**Model**					
			decoupled		semi-dynamic		fully dynamic	
	$\mathscr{C}$ $(\mathcal{O}(\cdot))$	T_{cp}[a]	$\mathscr{C}$ $(\mathcal{O}(\cdot))$	T_{cp}[a]	$\mathscr{C}$ $(\mathcal{O}(\cdot))$	T_{cp}[a]	$\mathscr{C}$ $(\mathcal{O}(\cdot))$	T_{cp}[a]
Sim.	$N\dfrac{T_{\mathrm{real}}}{T_{\mathrm{TTI}}}$	643.09 h	–	–	–	–	–	–
Calc. $\mu_i(\boldsymbol{y})$	–	–	$2^{N-1}N$	3.08 s	$2^{N-1}N$	3.08 s	$2^{N-1}N$	3.08 s
Var. aggr.	–	–	$2^N N^2$	2.60 s	$2^N N^2$	2.60 s	$2^N N^2$	2.60 s
Single Location[b]								
Startup Delay	$\dfrac{N_{\mathrm{F}}N_{\mathrm{G}}}{N_{\boldsymbol{u}}}$	0.08 s	LK	0.031 s	LKN_{G}^2	1.76 h	LKN_{G}^2	10.78 h
Starv. Prob.	$\dfrac{N_{\mathrm{F}}}{N_{\boldsymbol{u}}}$	0.0007 s	LK	0.0016 s	$K2^N N^2$	0.014 s	$K2^N N^2$	0.35 s
Entire Cell[b]								
Startup Delay	$N_{\mathrm{F}}N_{\mathrm{G}}$	60.60 s	$LKN_{\boldsymbol{u}}$	1.52 s	$LKN_{\mathrm{G}}^2N_{\boldsymbol{u}}$	5.61 h	$LKN_{\mathrm{G}}^2N_{\boldsymbol{u}}$	21.91 h
Starv. Prob.	N_{F}	0.03 s	$LKN_{\boldsymbol{u}}$	1.38 s	$K2^N N^2N_{\boldsymbol{u}}$	9.99 s	$K2^N N^2N_{\boldsymbol{u}}$	250.49 s

[a] The computations have been performed on the high performance computing (HPC) cluster of the center for information services and high performance computing (ZIH) at TU Dresden.

[b] The model calculations for the startup delay of single locations have been performed with a high resolution of $N_{\mathrm{G}} = 16\,000$ to resolve the steps in the CDF. For the entire cell, the grid number was reduced to $N_{\mathrm{G}} = 1000$ as integrating over the different locations destroys the steps anyways. The models have been evaluated at $N_{\boldsymbol{u}} = 711$ locations based on a $10\,\mathrm{m} \times 10\,\mathrm{m}$ grid.

Model

The complexity of the different models is dominated by determining the cell service rates $\mu_i(\boldsymbol{y})$, the state aggregation to determine state probabilities, and the (numerical) solution of the PDE and ODE systems. The service rates require for each of the N cells an integration over $N_{\boldsymbol{u}}$ discretized locations for each of the $L = 2^{N-1}$ neighboring interference scenarios. Then, the state aggregation method essentially involves the solution of a linear system of size $2^N \times 2^N$, which generally requires $\mathcal{O}\left(\left(2^N\right)^3\right)$ operations. However, since the system is sparse with only $N+2$ entries in each row, the complexity is approximated by $\mathcal{O}(2^N N^2)$.

All of the explicit formula's complexities are assumed to grow only linearly with the number of states LK and to be performed in negligible time, except for the matrix exponential. According to [ML78], computing the matrix exponential of an $n \times n$ matrix has a complexity of $\mathcal{O}(n^3)$. However, due to the sparsity of the involved matrices this bound is assumed to be coarse. Since the matrix exponential is based on matrix multiplications, a complexity of $\mathcal{O}(2^N N^2)$ is assumed as well.

Lastly, the complexity of the FVM, grows with the number of states LK and the number of time steps N_t and grid cells N_G. Due to the implicit Euler step, there is also the solution of a linear system involved, but this can be precomputed, because the matrix is constant, and is therefore negligible. The numbers N_t and N_G are not independent from each other, but have to meet the so called Courant–Friedrichs–Lewy (CFL) condition [LeV07]

$$\frac{\left(\max v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}\right)\Delta t}{\Delta q} \leq 1. \tag{3.84}$$

This essentially means that a higher resolution in space requires also smaller time steps[8]. Thus, the complexity grows quadratically in N_G.

Even though the overall operations may be comparable, the semi-dynamic model has advantages over the fully dynamic model by handling smaller systems apart from potential parallelization. This is due to more efficient storage usage and matrix operations, when dealing with smaller systems.

[8] Inspecting the principle of FVMs, this refers to the fact that a conservation law is applied to each of the discretized cells. To ensure correctness, it has to be guaranteed that matter (or concentration, probability, etc.) cannot move more than one cell per time step, which leads to the CFL condition.

Comparison

As can be seen in Table 3.3, generating the simulation data required $643.09\,\mathrm{h}$ (almost 27 days) of computation time, which was only feasible by dividing it into smaller jobs, executed in parallel thanks to HPC. In particular, when it comes to the evaluation of entire parameter sets, a lot of computing resources are required. Once the simulation data is available, the statistics of interest can be calculated in reasonable time. Only determining the CDF for the entire cell, which is a statistic of about 7.5 million flows took around $1\,\mathrm{min}$ of computation time. This analysis excludes the time that was needed to write the simulation data of the single jobs and to read it again to create the statistics, as this could have been done directly, if it would have been one single simulation.

In contrast, all models have to do the same calculation to obtain the state probabilities and service rates in advance, which requires less then six seconds in total. The decoupled model does not require any expensive numerical method, such that the startup delay distribution and the starvation probability can be approximated very fast. Only when it comes to evaluation at all of the $N_{\boldsymbol{u}}$ locations in the cell, more than one second is needed.

The other two models are also very fast in calculating the starvation probability. Since only explicit formulas are being calculated, the only expensive part is the matrix exponential and simulation time can be outperformed by several orders of magnitude. However, performing the numerical FVM is relatively expensive. For the single location statistics, a relatively high grid number N_{G} was used, such that the typical steps of the CDF (see Fig. 3.10) can be better resolved, leading to $1.76\,\mathrm{h}$ and $10.78\,\mathrm{h}$ for the semi-dynamic and the fully dynamic model, respectively. As this fine resolution is not necessary for the statistics over the entire cell, since the steps vanish through spatial integration, the resolution is reduced to $N_{\mathrm{G}} = 1000$ in this case, which still renders good approximations and requires $5.61\,\mathrm{h}$ and $21.91\,\mathrm{h}$ of computation time, respectively.

In summary, significant gains in computation time can be achieved by the models. The computation time for the starvation probability outperforms simulation times by several orders of magnitude. The calculation of the startup delay distribution is significantly faster as well, but still requires hours. However, this issue could be treated by more efficient numerical solvers. For instance, the underlying grid could be designed adaptively, such that it is very fine around the discontinuities, which is left for further studies.

3.5 Summary and Conclusion

In this chapter, modeling approaches for cellular wireless networks were proposed. The work is based on existing work for the DL, but extends the state of the art by incorporating interference dynamics, proposing numerical methods for the underlying PDE systems, as well as approaches for the UL direction. For the exemplary eMBB use case of video streaming, user and application-centric QoE metrics were derived and the model was validated by comparison with extensive system-level simulations. As a result, the influence of multi-cellular dynamics has been identified as crucial and, thus, should be taken into account for network analysis.

The work provides a powerful tool for network performance evaluation. Results could be transferred to other eMBB use cases. Depending on the particular interest, the DL model provides significant performance advantages in terms of computation time, compared with simulations. Even without performance gains, such models are of theoretical interest. In particular, the UL model has rather limited practical relevance in the current status. However, new numerical and analytical methods or appropriate simplifications may exploit the potential of the described models.

Performance evaluation frameworks enable insights how KPIs of interest depend on configurable parameters. This can help operators and service providers in configuring their systems. It also helps in understanding the relationships between parameters and KPIs and thereby in designing and testing optimization algorithms. Especially with regard to envisioned 5G applications, it is concluded that the focus should not be restricted on network-centric QoS metrics, but should include application-centric metrics that are more tailored to application-specific requirements.

There is still great potential for future studies. For instance, the described models may be extended in various aspects, such as introducing user mobility and thereby adding another degree of dynamics. Approaches for more realistic video traffic models or for VBR could be transferred to the model with interference and flow dynamics proposed in this work. Alternatively, the approach could be transferred to completely different applications. Finally, new numerical or analytical approaches may be applied to evaluate the derived models. For instance, the efficiency of the FVM could be greatly improved by making the underlying grid adaptive and, thus, reducing the necessary grid points.

4 Modeling the Access Network

For the realization of URLLC, the latency analysis cannot be restricted on the wireless access only. Instead, the entire path from a device to a (edge) server or to another device (and potentially back) has to be considered. This thesis focuses on the former, i. e., communication between a device and a server. Therein, the aim of the modeling is to determine the E2E latency distribution, which comprises not only the delay on the air interface, but also delays introduced by additional network elements between the BS and the server, where an application is running. It should be highlighted that by obtaining the E2E latency distribution, not only bounds or first and second moments of the latency will be provided, as in comparable work (e. g., [BBP04; Bon04b; Jar+11; Mah+14]), but also more relevant metrics for URLLC, such as high percentiles through the derivation of the entire CDF.

This chapter starts with introducing exemplary 5G architectures, which are the subject of investigation by mapping them to queuing networks. Modeling approaches for general homogeneous traffic (cf. Section 4.3) as well as for prioritized heterogeneous traffic (cf. Section 4.4) are proposed to be flexible with regard to different single or multiple applications. Although the proposed model can be applied to assess E2E latency in networks, not all aspects can be evaluated within the scope of this thesis. Hence, open research questions are summarized in a separate section (cf. Section 4.5). Again, the presented modeling approaches are put into perspective by comparing computational complexity and effort with simulations.

4.1 Scenarios

Fig. 4.1 shows a standard-compliant 5G architecture, as proposed in our work [Sch+19b]. It comprises an LTE as well as an NR RAN. UEs, including machines, may be connected to one or multiple antenna sites of one or even both RATs,

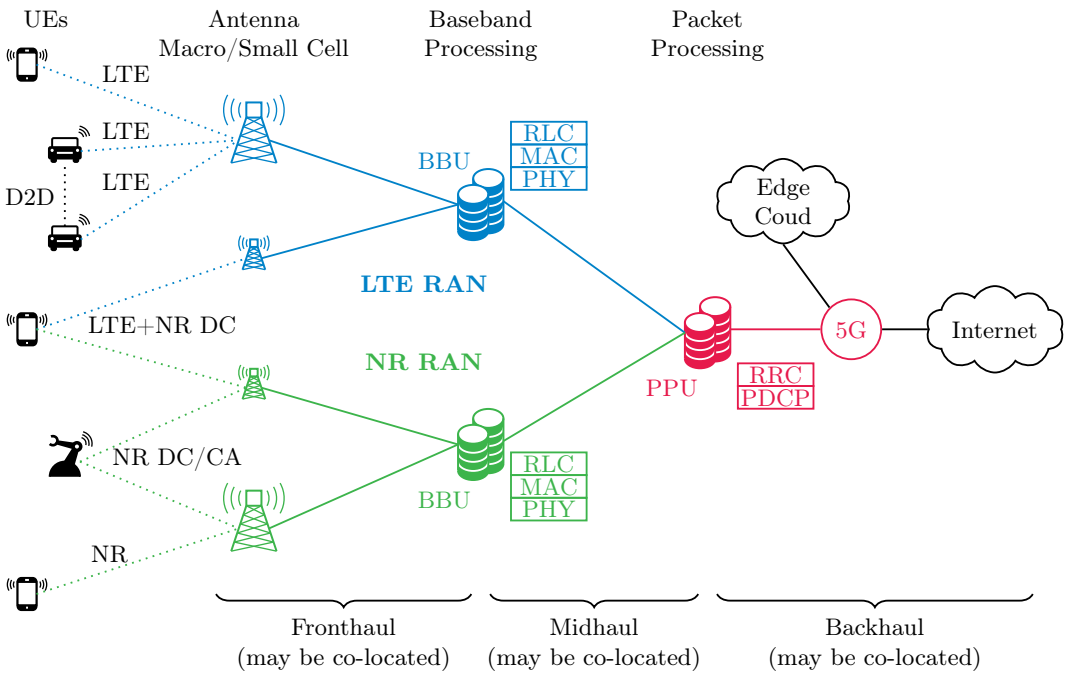
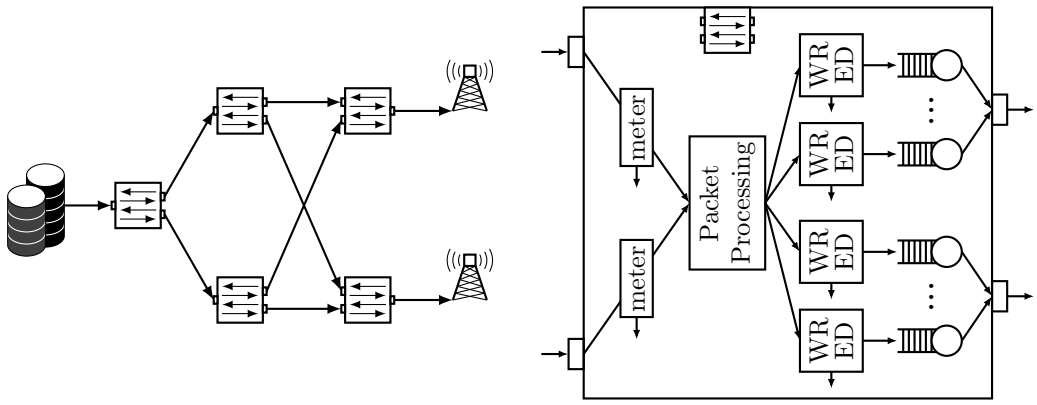


Fig. 4.1. Network architecture proposed in our work [Sch+19b] as a result of the collaborative research project *fast wireless* that is a part of the research cluster *fast - fast actuators sensors and transceivers* [Ell+16]. It comprises LTE as well as NR nodes.

where antennas may belong to small or macro cells. Furthermore, a direct communication, i. e., device-to-device (D2D), is foreseen, which is expected to be particularly beneficial for vehicle-to-vehicle (V2V) communication, but not in the scope of this thesis. The type of the connection depends on the application-specific requirements and the availability of the respective technologies. NR can provide low latency, but during its roll-out, wide coverage is mainly ensured by the already deployed LTE sites. Reliability can be achieved through MC concepts, which are so far standardized as dual connectivity (DC) [3GP13a] and carrier aggregation (CA) [3GP13b].

The antenna sites, or remote radio units (RRUs), are connected via the fronthaul to an LTE or NR base band unit (BBU), which is responsible for the following protocol layers: physical layer (PHY), medium access control (MAC), and radio link control (RLC). The BBU is then connected via the midhaul to a common packet processing unit (PPU), containing radio resource control (RRC) and packet data convergence protocol (PDCP) layer. The PPU is in turn connected via the backhaul to (edge) clouds and the Internet.

Each of the mentioned nodes between (and including) RRUs and PPUs can also be collocated, which is favored for low-latency applications, thanks to low



(a) Scenario (b) Ethernet switch model

Fig. 4.2. General RAN Scenario. (a) RAN architecture consisting of an application server, a network of Ethernet switches, and BSs. (b) Schematic illustration of an Ethernet switch structure.

propagation delays. However, the sketched functional split is part of a trend towards cloud-RAN or centralized-RAN (C-RAN) [Cha+19], which is expected to have advantages in terms of capital expenditure (CAPEX), operational expenditure (OPEX), and resource efficiency. Also, SON as well as network management and orchestration (MANO) may benefit from centralized entities.

4.1.1 General E2E Scenario

For the further analysis, a more general RAN scenario from a more macroscopic perspective, as depicted in Fig. 4.2a, shall be considered. The scenario consists of BSs and an application server with a network of Ethernet switches in between. Within such a scenario, the E2E latency is of particular interest, because a realistic assessment facilitates the virtual or physical deployment of network functions or (edge) cloud servers, with respect to the underlying latency constraints. Appropriate models can serve as good performance estimators. Thus, they can support finding reasonable decisions for dynamic allocation (or virtual deployment) of resources in a dynamic scenario. As the radio access was already discussed in Chapter 3, the focus is now shifted towards the network behind the radio interface.

4.1.2 Ethernet Switch Model

A simplified, schematic structure of an Ethernet switch is shown in Fig. 4.2b. It consists of several input ports, each one equipped with a metering unit, a packet

processing unit, and several output ports, each supporting weighted random early discard (WRED) and multiple queues. Of course, one input and one output port can refer to the same physical port at the switch, providing incoming and outgoing traffic from and to another device.

The metering units measure the incoming traffic and can drop any packets that violates pre-defined service level agreements (SLAs) [Kho+18] by exceeding the allowed traffic amount. SLAs between network operators and tenants define application requirements on both sides, i. e., what the network (slice) has to provide and what the application has to fulfill. Thus, one of these agreements could contain a maximum amount of traffic, for which the SLAs can be fulfilled. To take the metering into account for the modeling, arrival processes are assumed to fulfill the SLAs from application-side and, thus, do not violate the allowed traffic.

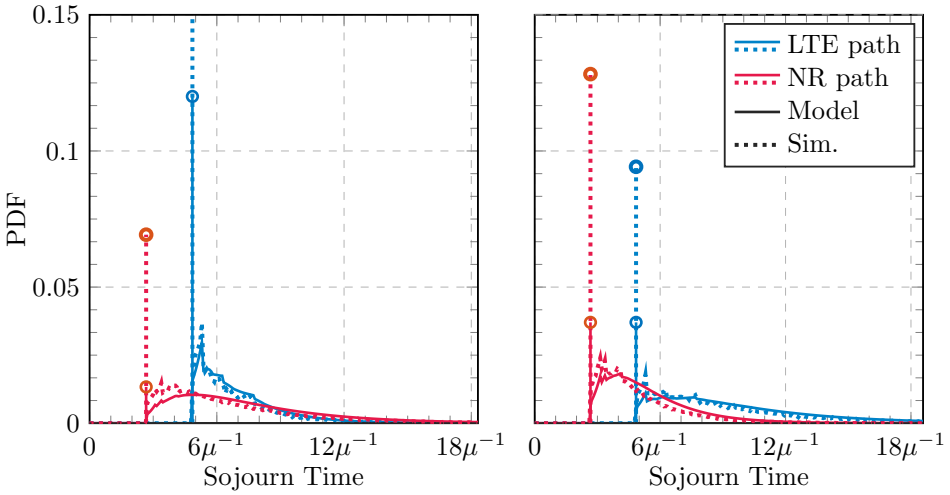
Packets are then forwarded to the packet processing, that essentially determines the destined output port and forwards the packets accordingly. Switches are dimensioned in a way that there is no queuing and the packet processing can be performed with a constant, small delay.

Before arriving in one of the output queues, WRED [FJ93; Nag18] is applied to avoid congestion. Here, packets are randomly dropped, depending on predefined weights, thresholds, and drop probabilities, as well as the instantaneous queue utilization. Modeling this feature is not in the scope of this work and may be pursued in future studies.

Finally, packets are queued in the output buffers of each port. Every port is equipped with several queues, such that packets can be sorted according to their traffic class and/or priority. For instance, traffic from different slices could be separated in dedicated output queues. In Section 4.4, different scheduling policies of priority queues will be integrated into the model.

4.2 Queuing Network

To analyze network performance, the architectures of the previous sections will be mapped onto a queuing network. When each of the network elements in Fig. 4.1 is mapped onto an $\mathrm{M/D/1}$ queue Q_i, reflecting Poisson arrivals and fixed TTI slots, interesting insights can be obtained already. The results were also presented in our work [Sch+19b].



(a) Equal load (b) Doubled load for NR

Fig. 4.3. E2E latency in the exemplary 5G architecture from Fig. 4.1. Both plots show the one-way sojourn time distribution from UE to edge cloud. The x-axes are normalized to the service rate of the entire NR branch. (a) Both technologies, LTE and NR, serve the same load. (b) NR carries twice the load of LTE.

Since NR provides a new numerology with potentially higher SCS (sub carrier spacing, cf. Section 1.1.2) [3GP18a] allowing shorter slot lengths, the service rate in the NR nodes is assumed to be higher than the LTE node by a factor k_{SCS}, i. e., $\mu_{\mathrm{NR}} = k_{\mathrm{SCS}}\mu_{\mathrm{LTE}}$. To have a fair comparison, the traffic being forwarded to the NR nodes is scaled by this factor as well, such that all nodes experience the same load ($\rho_{\mathrm{NR}} = \rho_{\mathrm{LTE}}$) and $\lambda_{\mathrm{NR}} = k_{\mathrm{SCS}}\lambda_{\mathrm{LTE}}$ holds. The packet processing is dimensioned accordingly with $\mu_{\mathrm{PP}} = \mu_{\mathrm{NR}} + \mu_{\mathrm{LTE}}$.

Fig. 4.3 shows model (Eq. 2.12) and simulation results for an SCS factor of $k_{\mathrm{SCS}} = 2$, reflecting the transition from an LTE SCS of $15\,\mathrm{kHz}$ to the NR option of $30\,\mathrm{kHz}$. Apart from the impulses at the lower ends of the PDFs, the model approximates the simulations well. The impulses refer to packets, which experienced no waiting in the network. Their deviation mainly stems from the fact that the network is not sufficiently dense to fulfill Assumption 2.2 (cf. Fig. 2.6). Here, the mutual dependence has the strongest impact on packets observing empty queues, since then it is likely to observe an empty waiting line at the next queue as well in sparse networks.

Interestingly, shortening the TTI length is not only beneficial for the minimal achievable latency, but also leads to a more compact PDF and thereby having a better distribution tail, as it can be seen in Fig. 4.3a. For subfigure 4.3b, the traffic routed through the NR branch is doubled again, i. e., $\lambda_{\mathrm{NR}} = 2k_{\mathrm{SCS}}\lambda_{\mathrm{LTE}}$

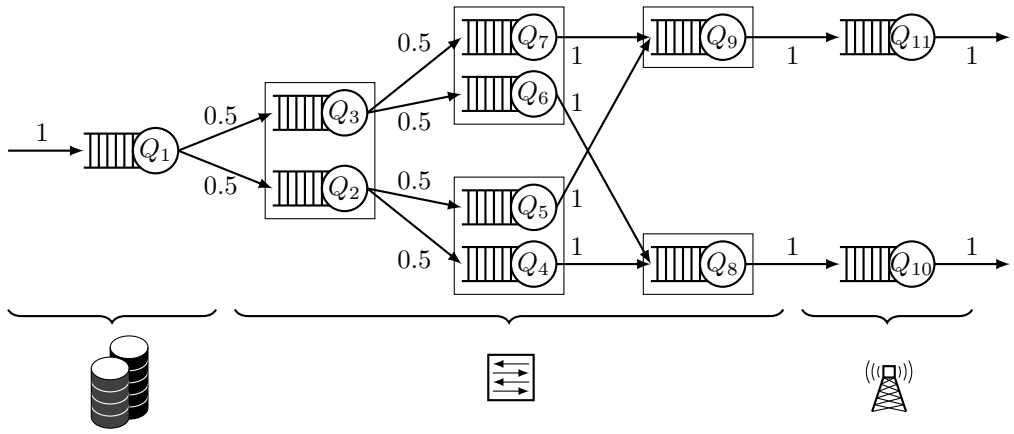


Fig. 4.4. The scenario of Fig. 4.2 translated into a queuing network. All switches are indicated by rectangular boxes. The ones with two outputs have been substituted by two queues.

or $\rho_{\mathrm{NR}} = 2\rho_{\mathrm{LTE}}$. Even though NR carries twice the load in this case, the latency performance is still much better in the lower percentiles, whereas the distribution tails of both branches coincide.

It should be noted that the model is kept very simple in this study and does not include any technical or implementation detail, such as processing or propagation delays. Even though they could be easily integrated via the variables D_i for a specific implementation, they are expected to shift the PDFs by constant delays. Thus, it is already interesting to investigate the impact achieved by queuing effects only. It also shows how strategies like traffic offloading can reshape latency PDFs, such that high percentiles performance can be improved by degrading lower percentiles.

The focus is now placed to the scenario again which was introduced in Section 4.1.1. The idea is that each network element will be modeled as one or multiple queues. By mapping each BS to one queue, and each of the switches to one queue per output, the architecture from Fig. 4.2 translates to the queuing network depicted in Fig. 4.4 for the DL.

To take also additional delays apart from queuing at each node into account, a new RV D_i can be introduced for each queue Q_i. For instance, this RV can comprise latency due to the packet processing in the switches or propagation delays at the links. It can accommodate deterministic as well as random effects, but in general these delays are assumed to be mutually independent. The quantity D_i can be taken into account by modifying Eq. (2.5) to

$$\tilde{J}_i = W_i + S_i + D_i. \tag{4.1}$$

The RV $\tilde{J}_i$ denotes the latency at queue Q_i. Accordingly, Eq. (2.12) can be modified as follows for the latency along a path

$$f_{\tilde{J}_{\boldsymbol{q}}}(t) = \left[\left(\overset{\kappa}{\underset{i=1}{\circledast}} f_{W_{q_i}}(s) \right) * \left(\overset{\kappa}{\underset{i=1}{\circledast}} f_{D_{q_i}}(s) \right) * \mu_{\boldsymbol{q}} f_{S_0}\left(\mu_{\boldsymbol{q}} s\right) \right](t). \tag{4.2}$$

As the RV D_i is expected to account mainly for deterministic effects, i. e., having a dirac impuls as a PDF, the convolution would simplify in most of the cases to a shift of the PDF.

However, D_i is set to zero for all queues in the remainder of this thesis for two reasons. First, as mentioned, as most of these additional delays are expected to be deterministic, the impact is rather low, because it affects all paths equally, if there is no difference in the number of hops. Second, packet processing and propagation delays are considered to be small enough in a RAN scenario compared to queuing delays, such that they are negligible.

4.3 Homogeneous Traffic

As a first step, homogeneous traffic from a single application is considered. With the various traffic models from Section 2.1 in mind and thinking of different hardware implementations, the first challenge arises, when it comes to finding appropriate queuing models. In particular, this is challenging, when the aim is not only finding the first stochastic moments (i. e., mean and variance), but also the entire PDF of the waiting or sojourn time. Thus, in the subsequent section, a numerical method is introduced for systems with general independent inter-arrival and service times, i. e., $\mathrm{GI}/\mathrm{GI}/1$ queues. Furthermore, a known result for a low-complexity upper performance bound is reproduced, as it complements the framework.

4.3.1 Waiting Time Distribution in $\mathbf{GI}/\mathbf{GI}/1$ Queues

In this section, the objective is to derive the steady state distribution of the waiting time, i. e.,

$$W := \lim_{n \to \infty} W_n, \tag{4.3}$$

Algorithm 4.1: Waiting time distribution for $\mathrm{GI/GI/1}$.

input : $f_S, f_{-T}, \mathrm{tol}$
output : Waiting Time pdf f_W
`/* initialize: */`
$i \leftarrow 0$;
$f_W^{(0)} \leftarrow \delta_0$;
$\Delta^{(0)} \leftarrow \infty$;
$\Delta^{(-1)} \leftarrow \infty$;
$f_U \leftarrow$ `Convolve(`f_S, f_{-T}`)` ;
while $\Delta^{(i)} > \mathrm{tol}$ ***and*** $\Delta^{(i)} \leq \Delta^{(i-1)}$ **do**
  $i \leftarrow i + 1$;
  $f_W^{(i)} \leftarrow$ `Convolve(`$f_W^{(i-1)}, f_U$`)`;
  $f_W^{(i)} \leftarrow$ `MoveNegativePartToZero(`$f_W^{(i)}$`)`;
  $\Delta^{(i)} \leftarrow$ `Norm(`$f_W^{(i)} - f_W^{(i-1)}$`)`;
end

for $\mathrm{GI/GI/1}$ FIFO queues. The common approach for deriving the waiting time is to introduce the auxiliary RV U_n [Kle75; Asm03] by

$$U_n := S_n - T_{n+1}. \tag{4.4}$$

The variable U_n can be interpreted as the *breathing time* of the $(n+1)^{\text{th}}$ object, i. e., the service time of its predecessor minus the time it arrives later than the predecessor or, in other words, the additional waiting time compared to its predecessor, which can be negative, if it arrives much later. Since arrival and service are independent, the PDF of U_n can be obtained by the following convolution

$$f_U(t) = (f_S * f_{-T})(t) = \int_{-\infty}^{+\infty} f_S(\tau) f_T(\tau - t) \mathrm{d}\tau. \tag{4.5}$$

Thereby, the breathing times U_n inherit the i.i.d. property. Due to the explanation above, the waiting time of the $(n+1)^{\text{th}}$ object can be related to the one of the n^{th} object as follows

$$W_{n+1} = \max\{W_n + U_n, 0\}, \tag{4.6}$$

which is known as *Lindley's equation*. Each object has to wait for the same time as its predecessor plus the additional waiting or time saved U_n. As the waiting time cannot be negative, the maximum operation is performed, which reflects the situation when T_{n+1} is sufficiently large, such that the current object arrives

to an empty system. Lindley has shown that the distribution of W_n converges, if the following condition holds [Lin52]

$$\mathbb{E}[U] < 0. \tag{4.7}$$

The condition essentially says that service is on average shorter than the time between arrivals, which corresponds to the condition $\rho = \frac{\lambda}{\mu} < 1$ for the arrival and service rates. Lindley's equation Eq. (4.6) can also be formulated for the respective PDFs as the functional equation

$$f_{W_{n+1}} = \phi\left(f_{W_n} * f_{U_n}\right). \tag{4.8}$$

Here, the operator ϕ is the equivalent to the maximum operation of Eq. (4.6) in the PDF domain. It pulls the negative part of a function f to a weighted Dirac impulse δ_0 at zero, i. e.,

$$\phi f := \chi_{[0,\infty)} \cdot f + \left(\int_{(-\infty,0)} f(t)\mathrm{d}t\right) \cdot \delta_0, \tag{4.9}$$

with χ_A being the indicator function of a set A. The operator can also be defined for a CDF F, which results in setting the negative part to zero

$$\Phi F := \chi_{[0,\infty)} \cdot F. \tag{4.10}$$

By taking the limit $n \to \infty$, Eq. (4.8) becomes a fixed point formulation

$$f_W = \phi\left(f_W * f_U\right) =: \varphi_U(f_W), \tag{4.11}$$

for the waiting time in equilibrium. This constitutes a fixed point equation, i. e., an equation of the form $x = \varphi(x)$, which needs to be solved for x. In this particular case, the PDF f_W is the unknown of a functional equation.

Lindley's equation is hard to solve for the general case. There exist approaches for specific cases that usually use Laplace transforms, which are hard to be applied to a general case, if not performed numerically, which is also not trivial. For instance, the authors in [VA07] provide a solution for a slightly different, so called Lindley-Type equation (with a negative sign of W_n in Eq. (4.6)) for exponentially distributed T, polynomially distributed S and finite support, which can also be used for approximations of the general case. Another common approach uses Wiener-Hopf decomposition [Pra74], which transforms and splits the equation in a pair of complex functions in the upper and lower halves of

the complex plain, respectively. However, this method is also not suitable for a framework that should be able to handle general systems.

Instead, a fixed point iteration to solve Eq. (4.11) is proposed in this thesis and in our work [Sch+19a]. The term fixed point, should not bring any confusion as points refer to functions here. The solver starts with an initial PDF $f_W^{(0)} = \delta_0$ on which Eq. (4.11) is applied as long as it converges. The L^2 norm is used to define the following metrics for two RVs X and Y

$$d_f(X,Y) := \|f_X - f_Y\|_{L^2} = \left(\int_{-\infty}^{+\infty} |f_X(\tau) - f_Y(\tau)|^2 \,\mathrm{d}\tau\right)^{\frac{1}{2}}, \tag{4.12}$$

$$d_F(X,Y) := \|F_X - F_Y\|_{L^2} = \left(\int_{-\infty}^{+\infty} |F_X(\tau) - F_Y(\tau)|^2 \,\mathrm{d}\tau\right)^{\frac{1}{2}}. \tag{4.13}$$

With the metric $d_f(\cdot,\cdot)$, the difference $\Delta^{(i)}$ of two consecutive iterations is measured and taken as an exit criterion. The numerical method stops if $\Delta^{(i)}$ either falls below a predefined tolerance threshold tol or if $\Delta^{(i)}$ grows, due to numerical instabilities for small increments. The entire approach is summarized in Algorithm 4.1.

The algorithm can be interpreted as follows. Lindley's equation is evaluated over and over, what is similar to simulating the system. However, in the algorithm the PDF is used, which evaluates all possible constellations at once, but weighted according to their probabilities.

The convergence behavior is stated in the following proposition.

Proposition 4.1 (Convergence of Algorithm 4.1 for $\mathrm{GI/GI/1}$ Queues.)**.** *Let S_n and T_n be RVs on $[0,\infty)$ for $n \in \mathbb{N}$, such that both series (S_n) and (T_n) are i.i.d.. If Eq.* (4.7) *holds for $U_n = S_n - T_n$, $n \in \mathbb{N}$ then the series of RVs (W_n), defined by the fixed-point iteration*

$$W_0 := 0, \tag{4.14}$$

$$W_{n+1} := \varphi_U(W_n) := \pi(W_n + U_n), \tag{4.15}$$

or equivalently defined by their PDFs,

$$f_{W_{n+1}} := \phi(f_{W_n} * f_{U_n}), \tag{4.16}$$

or their CDFs,

$$F_{W_{n+1}} := \Phi(F_{W_n} * f_{U_n}), \tag{4.17}$$

respectively, in Algorithm 4.1 converges in distribution to the unique solution W of Lindley's equation Eq. (4.6).

Proof. See Appendix A.3. □

4.3.2 An Upper Bound for $\mathbf{GI/GI/1}$ Systems

Using the result from [Kin64], the waiting time can be bounded as follows

$$\mathbb{P}\left[W > t\right] \leq e^{-\theta_0 t} \tag{4.18}$$

with

$$\theta_0 = \sup\left\{\theta \in \mathbb{R}_{>0} | M_U(\theta) < 1\right\}, \tag{4.19}$$

where M_U is the moment generating function (MGF) of U

$$M_U(\theta) = \mathbb{E}\left[e^{\theta U}\right]. \tag{4.20}$$

Eq. (4.18) provides an upper performance bound that is relatively simple to be calculated (depending on the distributions of T and S maybe even analytically) and that provides bounds for high percentiles as well. The bound can be directly applied for the CDF

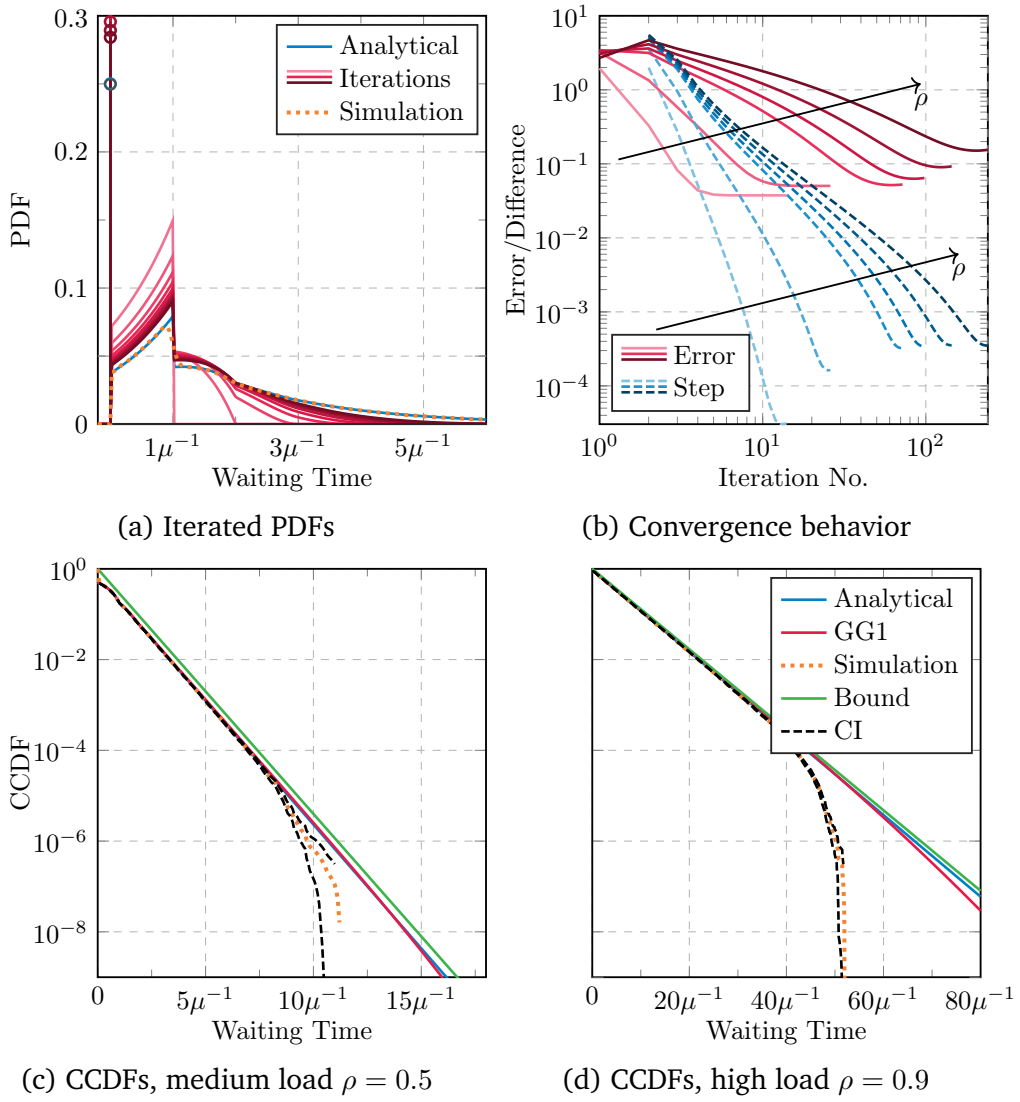
$$F_W(t) = \mathbb{P}\left[W \leq t\right] = 1 - \mathbb{P}\left[W > t\right] \geq 1 - e^{-\theta_0 t}. \tag{4.21}$$

4.3.3 Numerical Validation

Algorithm 4.1 was tested with $\mathrm{M/D/1}$ queues for two reasons. First, $\mathrm{M/D/1}$ constitutes one of the URLLC traffic models (cf. Section 2.1.1) and second, because for $\mathrm{M/D/1}$ queues, the waiting time is known to be distributed as

$$F_W(t) = \pi_0 \sum_{k=0}^{m} \frac{\left\{-\lambda\left(t - k\bar{S}\right)\right\}^k}{k!} e^{-\lambda\left(t-k\bar{S}\right)} \quad \text{for } m\bar{S} \leq t < (m+1)\bar{S} \tag{4.22}$$

with the deterministic service time $\bar{S} = \mu^{-1}$ according to [Cro32]. However, this formula suffers from numerical instabilities, as it contains alternating terms that



(a) Iterated PDFs

(b) Convergence behavior

(c) CCDFs, medium load $\rho = 0.5$

(d) CCDFs, high load $\rho = 0.9$

Fig. 4.5. Performance of the proposed numerical approach for approximating the waiting time of GI/GI/1 queues. (a) Illustration of the converging PDFs. (b) Error and difference of consecutive iterations for different network loads. (c), (d) CCDFs of the analytical solution, the algorithm, the simulation including 95 % confidence intervals (CIs), and the upper bound for GI/GI/1 queues for medium and high load, respectively.

are much larger then their sum. Thus, the following numerically more stable version of this equation provided by [Fra01, Eq. (11)] is used in this thesis.

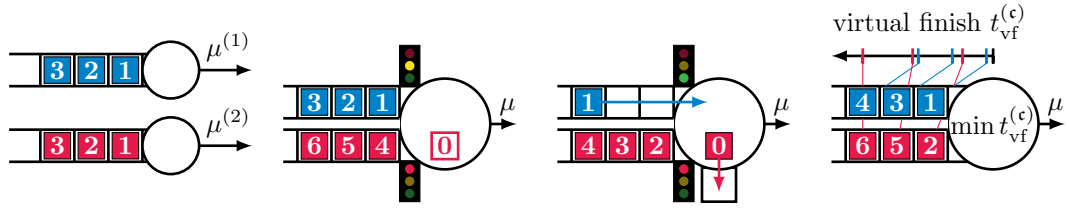
$$F_W(t) = e^{-\lambda\left(m\bar{S}-t\right)} \sum_{k=0}^{m} \left(\sum_{j=0}^{m-k} \pi_j \right) \frac{\left\{\lambda\left(m\bar{S}-t\right)\right\}^k}{k!} \quad \text{for } m\bar{S} \leq t < (m+1)\bar{S} \tag{4.23}$$

This formula needs the state probabilities π_i of an $\mathrm{M/D/1}$ queue, which can be obtained by a series expansion [Nak05, Eq. (2.4)], a fast Fourier transformation (FFT) method [Tij95], or an geometric tail approach [Tij95].

The experiment was conducted for an arrival and service rate of $\lambda = 0.15$ and $\mu = 0.2$, respectively, corresponding to a load of $\rho = 0.75$. The results are depicted in Fig. 4.5. Subfigure 4.5a shows how the iterated PDFs, depicted in different shades of red (from bright to dark), converge to the analytical solution drawn as a black solid line. The plot also shows curves from simulation (dashed line) for comparison. As a result, iterations with a higher index are almost indistinguishable from the analytical solution. The iterative solutions even manage to resolve the sharp edges of the analytical solution remarkably well. In contrast, simulation results fail in resolving discontinuities due to the kernel estimation of the PDF.

The CDFs of the same distributions are shown in subfigures 4.5c and d for medium and high load, respectively. They include the upper performance bound from Eq. (4.21), which turns out to be very tight for this scenario. Furthermore, it can be observed, how the empirical CDF of simulation results drifts away from the analytical and modeled curve at the tail. The figure also contains the estimated $95\,\%$ confidence interval (based on Greenwood's formula [Gre26], [KM58, Eq. (6d)]) of the empirical CDF, which is spreading more and more towards the tail, indicating less reliable results. In addition, the model has significant advantages in terms of computation complexity, as shown in Section 4.6.4.

In plot 4.5b, it is shown how the approximation error and the differences between consecutive iterations (step) behave with respect to the number of iterations for different values of the network load $\rho \in \{0.25, 0.5, 0.75, 0.8, 0.85, 0.9\}$. As expected (cf. the explanation why the algorithm works in Section 4.3.1), the convergence speed decreases with network load. However, in this work moderate load conditions are of higher interest, because URLLC is considered to be infeasible for high loads. Results for other queuing systems are provided in Appendix A.2.



(a) DR (b) PQwop (c) PQwp (d) WFQ

Fig. 4.6. Illustration of the different scheduling policies with numbers indicating the order of service. (a) Resources are separated. (b) Class $\mathfrak{c} = 1$ traffic only has to wait if a class $\mathfrak{c} = 2$ packet is still in service. (c) A class $\mathfrak{c} = 2$ service will be paused, when a high priority packet arrives. (d) The scheduling is based on a virtual finish time with different rates.

4.4 Heterogeneous Traffic

Network slices are introduced to handle traffic from multiple applications and hence different characteristics and requirements. In particular, traffic can have different priorities, such as URLLC being more important than eMBB. As described in Section 4.1.2, network hardware is already prepared by providing multiple output queues. However, the question remains how to schedule packets from different output queues onto the same link and how to integrate this scheduling into the model. In the following subsections, four different scheduling policies are explained and modeled (cf. our work [Sch+19a]). The proposed models will also be validated by simulation results.

4.4.1 Modeling of different Scheduling Policies

Let the network be confronted with traffic classes $\mathfrak{c} \in \mathcal{C}$. In the previous section, Fig. 4.4 illustrated the queuing network for a single class of traffic. When multiple classes are present, which are handled by separate queues, each queue Q_i in Fig. 4.4 has to be substituted by queues $Q_i^{(\mathfrak{c})}$ for each class. Herein, each queue may have different characteristics reflecting the different class traffic characteristics. Variables that refer to a certain class are indicated with a superscript $(\mathfrak{c})$. When the superscript o is added, the entire node comprising all classes and resources is meant.

In this thesis, two classes $\mathcal{C} = \{1, 2\}$ are considered. However, the models could be generalized to support more classes. Without loss of generality, class $\mathfrak{c} = 1$ is assumed to be the class with higher (or equal) priority.

Dedicated Resources

The first and most simple scheduling policy is called dedicated resources (DR). When each slice has dedicated resources, they do not interfere with each other, such that they can be modeled separately. Therefore, let slice one have the share $w_1 \in (0,1)$ of the resources and slice two $w_2 = 1 - w_1$, respectively. With this, the service times are scaled $S^{(\mathfrak{c})} = \frac{S^{\mathrm{o}}}{w_{\mathfrak{c}}}$, such that the effective rate of each class becomes $\mu^{(\mathfrak{c})} = w_{\mathfrak{c}}\mu^{\mathrm{o}}$ for queue $Q^{(\mathfrak{c})}$ and the waiting time can be assessed by analyzing separated queues. However, if one class is idle, the free resources cannot be used by the other class, because they are exclusively dedicated, leading to an inefficient scheduling.

Priority Queuing with preemption

Another approach is priority queuing. Here, the lower priority class can only transmit, when there is no traffic of higher priority. It can be implemented as priority queuing with preemption (PQwp) or priority queuing without preemption (PQwop). Preemption means that traffic from high priority classes can even interrupt transmissions of low priority, which will be continued, when the capacity is available again. It is a concept of time-sensitive network (TSN) that allows high-priority packets in the presence of longer low priority packets. The additional delay caused by preemption is comparatively small [TE16].

Since in PQwp the high priority traffic is always scheduled, it is not interfered by the other class and can be treated as a separated queue, with the inter-arrival time distribution $A^{(1)}$ of class one, but with all resources available and, hence, the service of the entire node B^{o}.

The waiting time of the low-priority class can then be expressed as follows

$$W^{(2)}_{\mathrm{PQwp}} = W^{\mathrm{o}} + \hat{W}^{(1)}\left(W^{(2)}_{\mathrm{PQwp}} + S^{(2)}\right). \tag{4.24}$$

Here, W^{o} denotes the waiting time of the entire node, if both classes were served jointly without any priorities, because the considered low priority packet has to wait for all packets that arrived before, no matter of which class they are and in which order they are served. While waiting, additional packets of the high priority class may arrive, which increase the waiting time of the considered packet. To take this into account, the function $\hat{W}^{(1)}(T_W)$ is introduced which denotes the additional waiting time due to class $\mathfrak{c} = 1$ packets arriving within time T_W. In this case, T_W consists of the own waiting and service time (due to potential preemption), yielding a fixed point formulation again that can be

solved by fixed point iteration. An expression for $\hat{W}^{(1)}(T_W)$ is derived in a subsequent section.

Priority Queuing without preemption

The modeling of PQwop is very similar to PQwp. However, since the high priority traffic may have to wait, when a lower priority packet is still in service, the waiting time distribution has to be modified accordingly. It can be implemented by adjusting the probability that a high priority packet has to wait. With the probability π_0^{o} of the entire node being empty, there is no waiting. Otherwise it has to wait for the waiting time $W_{x>0}^{(1)}$ conditioned on the queue not being empty. This results in the mixed RV

$$W_{\mathrm{PQwop}}^{(1)} = \begin{cases} \delta_0, & \text{w.p. } \pi_0^{\mathrm{o}}, \\ W_{x>0}^{(1)}, & \text{w.p. } \bar{\pi}_0^{\mathrm{o}}. \end{cases} \tag{4.25}$$

The waiting time of the second class is similar to Eq. (4.24). However, now for the additional waiting time due to high priority arrivals, the service time is not taken into account, because once service is started, additional arrivals have no influence anymore.

$$W_{\mathrm{PQwop}}^{(2)} = W^{\mathrm{o}} + \hat{W}^{(1)}\left(W_{\mathrm{PQwop}}^{(2)}\right). \tag{4.26}$$

Derivation of the Additional Waiting Time for Priority Queuing

For both policies with priority queuing, PQwp and PQwop, an expression for the additional waiting time $\hat{W}(T_W)$ caused by arrivals within the random time T_W is required. The derivation involves the RVs $\mathcal{S}_k = \sum_{l=1}^{k} S_l$ and $\mathcal{T}_k = \sum_{l=1}^{k} T_l$ of the sum of k independent service times and inter-arrival times, respectively. With this, the number of arrivals $\mathcal{A}_t$ within a fixed time t, can be expressed as

$$\begin{aligned} \mathbb{P}\left[\mathcal{A}_t = k\right] &= \mathbb{P}\left[\mathcal{A}_t \geq k\right] - \mathbb{P}\left[\mathcal{A}_t \geq k-1\right], && (4.27) \\ &= F_{\mathcal{T}_k}(t) - F_{\mathcal{T}_{k-1}}(t), && (4.28) \end{aligned}$$

which already provides the desired expression for a fixed time t by summing the respective service times of the arrived objects

$$\hat{W}(t) = \sum_{k=0}^{\infty} \mathbb{P}\left[\mathcal{A}_t = k\right] \mathcal{S}_k. \tag{4.29}$$

Algorithm 4.2: Weighted fair queuing (WFQ).

input : $(w_{\mathfrak{c}}), (t^{(\mathfrak{c})}_{\mathrm{arr},n}), (t^{(\mathfrak{c})}_{\mathrm{dep},n})$
output: Scheduled class $\mathfrak{c}^*$
for $\mathfrak{c} \in \mathcal{C}$ **do**
  if $Q_i^{(\mathfrak{c})}$ *is empty* **then**
    Virtual Finish $t^{(\mathfrak{c})}_{\mathrm{vf},n} = \infty$;
  else
    $n \leftarrow$ index of the packet at the head of $Q_i^{(\mathfrak{c})}$;
    Virtual Rate $R_{\mathfrak{c}} \leftarrow w_{\mathfrak{c}}\mu^{\mathrm{o}}$;
    Virtual Start $t^{(\mathfrak{c})}_{\mathrm{vs},n} = \max\left\{t^{(\mathfrak{c})}_{\mathrm{arr},n}, t^{(\mathfrak{c})}_{\mathrm{vf},n-1}\right\}$;
    Virtual Finish $t^{(\mathfrak{c})}_{\mathrm{vf},n} = t^{(\mathfrak{c})}_{\mathrm{vs},n} + \frac{s_n^{(\mathfrak{c})}}{R_{\mathfrak{c}}}$;
  end
end
Scheduled class $\mathfrak{c}^* \leftarrow \mathrm{argmin}_{\mathfrak{c}}\left\{t^{(\mathfrak{c})}_{\mathrm{vf},n}\right\}$

For a random time T_W, the additional waiting time can then be obtained by integrating and weighting with the PDF f_{T_W} according to

$$\hat{W}(T_W) = \sum_{k=0}^{\infty}\left(\int_{-\infty}^{\infty} f_{T_W}(t)\mathbb{P}\left[\mathcal{A}_t = k\right]\mathrm{d}t\right)\mathcal{S}_k. \tag{4.30}$$

Here, the integral expresses the probability of having k arrivals within a random time T_W.

Since Eqs. (4.24) and (4.26) have the form $Z := \hat{W}(T_W) + T_W$, the addends are not independent from each other, and the convolution would lead to inaccurate results. This can be resolved by directly deriving the distribution of Z as follows

$$f_Z(\xi) = \frac{\mathrm{d}}{\mathrm{d}z}F_Z(\xi) = \mathbb{P}\left[W(T_W) + T_W \leq \xi\right], \tag{4.31}$$

$$= \frac{\mathrm{d}}{\mathrm{d}z}\left(\int_{-\infty}^{\infty} f_{T_W}(t)\sum_{k=0}^{\infty}\mathbb{P}\left[\mathcal{A}_t = k\right]\int_{-\infty}^{\xi-t} f_{\mathcal{S}_k}(\omega)\mathrm{d}\omega\mathrm{d}t\right), \tag{4.32}$$

$$= \sum_{k=0}^{\infty}\left(\left(f_{T_W}(t)\cdot\mathbb{P}\left[\mathcal{A}_t = k\right]\right) * f_{\mathcal{S}_k}\right)(\xi). \tag{4.33}$$

It should be noted that for the last line, the term $f_{T_W}(t)\cdot\mathbb{P}\left[\mathcal{A}_t = k\right]$ is considered as a function in t that is convolved with $f_{\mathcal{S}_k}$.

Weighted Fair Queuing

The fourth considered scheduling policy is called weighted fair queuing (WFQ) [PG93]. It can be categorized in between DR and the priority queuing, by performing a weighted RR and thus allowing prioritization of traffic. It is implemented as sketched in Algorithm 4.2. For each traffic class, a virtual rate $R_{\mathfrak{c}}$ is calculated, based on predefined weights $w_{\mathfrak{c}}$. A virtual start $t_{\mathrm{vf},n}^{(\mathfrak{c})}$ is assigned to each packet, based on its own arrival time $t_{\mathrm{arr},n}^{(\mathfrak{c})}$ and the virtual finish $t_{\mathrm{vf},n-1}^{(\mathfrak{c})}$ of its predecessor. Finally, from both, the virtual start $t_{\mathrm{vs},n}^{(\mathfrak{c})}$ and the virtual rate $R_{\mathfrak{c}}$, a virtual finish $t_{\mathrm{vf},n}^{(\mathfrak{c})}$ is derived. The scheduled packet is then the one with the earliest virtual finish.

Because it would be very cumbersome to put this algorithm into a stochastic analysis, the following approximation is conducted instead. First, two limiting cases will be studied. By setting the weights to the boundary values $(w_1, w_2) = (1, 0)$, class $\mathfrak{c} = 1$ will always be favored, because the virtual rate of class $\mathfrak{c} = 2$ becomes $R_2 = 0$ leading to a virtual finish $t_{\mathrm{vf},n}^{(2)} = \infty$. This way, WFQ degenerates to PQwop. In contrast, if the weights are set to the other boundary case $(w_1, w_2) = (0.5, 0.5)$ the scheduling ends up in regular RR and both classes experience the waiting time of the entire node W^{o}.

This suggests the approximations by the following mixtures of RVs that interpolate between the extreme cases according to the chosen weights $w_{\mathfrak{c}}$

$$W_{\mathrm{WFQ}}^{(1)} \cong \begin{cases} W_{\mathrm{PQwop}}^{(1)}, & \text{w.p. } 1 - 4(w_1 - 1)^2 \\ W^{\mathrm{o}}, & \text{w.p. } 4(w_1 - 1)^2 \end{cases}, \tag{4.34}$$

$$W_{\mathrm{WFQ}}^{(2)} \cong \begin{cases} W^{\mathrm{o}}, & \text{w.p. } 1 - 4w_2^2 \\ W_{\mathrm{PQwop}}^{(2)}, & \text{w.p. } 4w_2^2 \end{cases}. \tag{4.35}$$

Here, a quadratic interpolation was chosen, because the ratio $\frac{w_1}{w_2} = \frac{w_1}{1-w_1}$, which determines the prioritization, grows quadratically in w_1. This can be observed by the derivation

$$\frac{\mathrm{d}}{\mathrm{d}w_1}\left(\frac{w_1}{1-w_1}\right) = \frac{1}{(1-w_1)^2}. \tag{4.36}$$

This way, the approximation converges faster to PQwop for high weighting of class $\mathfrak{c} = 1$.

Approaches for more than two classes

Even though the modeling of more than two classes is out of scope of this thesis, approaches for the extensions shall be sketched shortly. The DR scheme can be easily extended, since the queues are treated separately and do not mutually interfere. For the priority queuing schemes, Eq. (4.26) can be adjusted as follows for an additional class $\mathfrak{c} = 3$

$$W^{(2)}_{\mathrm{PQwop}} = W^{(1)+(2)} + \hat{W}^{(1)}\left(W^{(2)}_{\mathrm{PQwp}}\right), \tag{4.37}$$

$$W^{(3)}_{\mathrm{PQwop}} = W^{\mathrm{o}} + \hat{W}^{(1)}\left(W^{(3)}_{\mathrm{PQwp}}\right) + \hat{W}^{(2)}\left(W^{(3)}_{\mathrm{PQwp}}\right). \tag{4.38}$$

Here, $W^{(1)+(2)}$ denotes the waiting time, occurring in a queue that would handle the first two classes jointly. Further classes can be added in the same way.

If WFQ scheduling is considered for three classes, one can consider three extreme cases:

(i) $(w_1, w_2, w_3) = (1, 0, 0)$: Class $\mathfrak{c} = 1$ is handled as priority, the others jointly with RR.

(ii) $(w_1, w_2, w_3) = \left(\frac{1}{2}, \frac{1}{2}, 0\right)$: Classes $\mathfrak{c} \in \{1, 2\}$ are modeled jointly as RR with modified π_0, the low priority class as $W^{(3)}_{\mathrm{PQwop}}$.

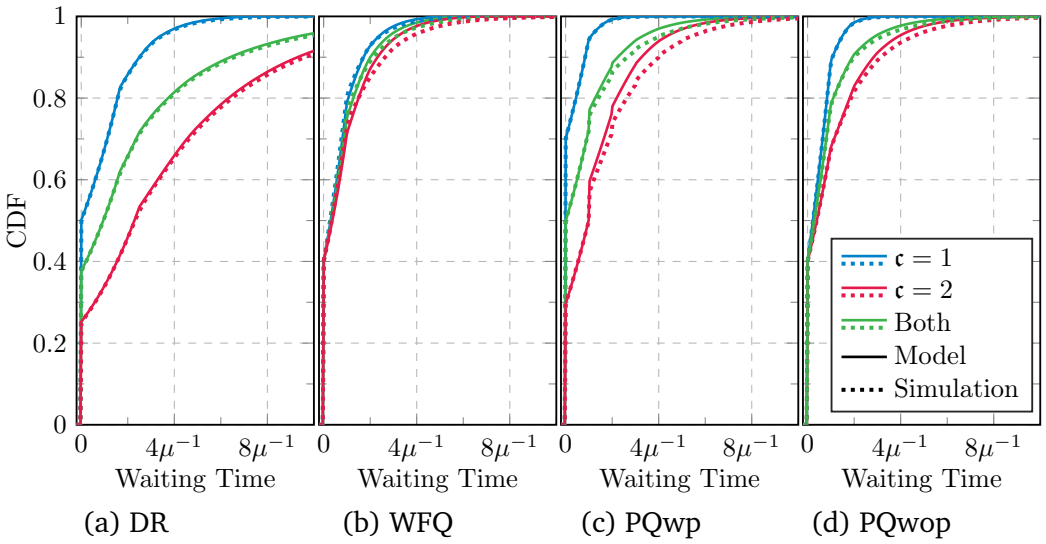
(iii) $(w_1, w_2, w_3) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$: All classes are modeled jointly with RR.

Using barycentric coordinates [Far02, Section 2.1.3] and quadratic Bezier triangles [Far02, Section 4.4] as the interpolation method, the waiting time distribution can then be approximated for the chosen weights (w_1, w_2, w_3). This approach would also be suitable for a higher number of considered classes, where the extreme cases would lead to the vertices of a $(|\mathcal{C}| - 1)$-dimensional convex simplex.

4.4.2 Numerical Validation

The accuracy of the models has been validated by comparison to the distributions generated by simulating a single node with two traffic classes. As in Section 4.3, both classes are assumed to follow $\mathrm{M/D/1}$ models. In contrast to the homogeneous case, no analytical results are available for the four schedulers that could be taken as a benchmark.

By setting the arrival rate for each class to $\alpha^{(\mathfrak{c})} = 0.06$ and the service rate of the entire node to $\mu^{\mathrm{o}} = 0.2$, the node is put into moderate overall load conditions



(a) DR (b) WFQ (c) PQwp (d) PQwop

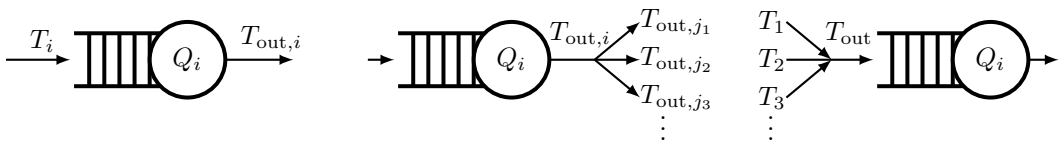
Fig. 4.7. Modeling performance for the different schedulers.

with $\rho^{\circ} = 0.6$. Here, the overall load cannot be set too high, since this may lead to instabilities due to high individual load for the low priority queues, especially in the DR case. For both schedulers with parameters, i. e., DR and WFQ, the weights are set to $(w_1, w_2) = (0.6, 0.4)$.

The results of the validation are depicted in Fig. 4.7. Here, the modeled and simulated CDFs are plotted for both classes and the entire node. It can be observed that the models approximate the simulation curves well for all four scheduling policies, especially for the high priority class, which is more important for URLLC. Furthermore, it can be seen that WFQ lies in between RR, which refers to the green curves in Fig. 4.7b, and PQwop, since those are the extreme cases. Thus, by setting the weights accordingly, WFQ can realize curves between RR and PQwop. By comparing the schedulers, it becomes evident that DR is the most inefficient with the worst performance for both classes due to the wasting of resources. The other approaches are trading off the performance of both classes against each other. For instance, PQwp shows the best performance for high priority traffic, but also the worst for low priority, among WFQ, PQwop, and PQwp.

4.5 Open Questions

In the previous sections, methods to obtain the distribution of the waiting time for queues with general arrivals and service as well as for heterogeneous traffic along with its scheduling are introduced. In fact, each queue also changes the



(a) Passing a node (b) Split traffic (c) Merge traffic

Fig. 4.8. Basic operations in a queuing network.

behavior of the traffic that is going through. Thus, even if the traffic processes at the input queues, generated by the application, is known (cf. Section 2.1), it is not clear how the traffic pattern looks like after passing the first queue. In other words, the arrival process has to be determined for the inner queues of the network.

Fig. 4.8 shows basic operations that have to be modeled for this purpose. If all of them can be treated, the network can be decomposed, similar to the approach in [Küh76], where the operations are performed for the first two stochastic moments. In the subsequent sections approaches are sketched for what needs to be done to obtain the full distribution of the process and thereby to complete the framework.

4.5.1 Passing a Node

When traffic goes through a queue Q_i the inter-departure times form the inter-arrival time T_{out} for a queue that would be located behind. As long as Q_i is busy, the inter-departure times are given by the service times S_i. In contrast, if Q_i is idle, the next departure happens after one inter-arrival time T_i. This results in the mixed RV

$$T_{\text{out}} = \begin{cases} T_i + S_i & \text{w.p. } \pi_{i,0} \\ S_i & \text{w.p. } 1 - \pi_{i,0} \end{cases}, \tag{4.39}$$

where $\pi_{i,0}$ denotes the probability of queue Q_i being empty.

4.5.2 Split Traffic

If the output of one queue Q_i is forwarded to multiple queues, the traffic has to be split. Assuming that the last object was forwarded to queue Q_j, the probability that the next one is forwarded to Q_j is p_{ij}, resulting in the inter-arrival time $T_{\text{out},i}^{(1)}$.

With the complementary probability $1-p_{ij}$ another queue was chosen and so the probability that the second object is the first one forwarded to Q_j is $(1-p_{ij})p_{ij}$ with an inter-arrival time of $T_{\text{out},i}^{(1)} + T_{\text{out},i}^{(2)}$. Continuing this thought leads to a mixed RV weighted by Bernoulli probabilities for choosing the particular branch after a certain number of departures:

$$T_{\text{out},j} = \begin{cases} T_{\text{out},i}^{(1)}, & \text{w.p. } p_{ij} \\ T_{\text{out},i}^{(1)} + T_{\text{out},i}^{(2)}, & \text{w.p. } (1-p_{ij})p_{ij} \\ T_{\text{out},i}^{(1)} + T_{\text{out},i}^{(2)} + T_{\text{out},i}^{(3)}, & \text{w.p. } (1-p_{ij})^2 p_{ij} \\ \quad \vdots & \quad \vdots \\ \sum_{l=1}^{k} T_{\text{out},i}^{(l)}, & \text{w.p. } (1-p_{ij})^{k-1} p_{ij} \\ \quad \vdots & \quad \vdots \end{cases} . \tag{4.40}$$

Here, the $T_{\text{out},i}^{(l)}$ denote independently drawn RVs of the departure distribution of queue Q_i.

4.5.3 Merge Traffic

Merging two or more arrival processes to obtain the resulting inter-arrival time distribution is a cumbersome task for general processes, as described in the following for the case of only two processes T_1 and T_2. Starting with an empty system, the first resulting inter-arrival time $T_{\text{out}}^{(1)}$ is the minimum of the first times of both input processes

$$T_{\text{out}}^{(1)} = \min\left\{T_1^{(1)}, T_2^{(1)}\right\}. \tag{4.41}$$

But inspecting the next inter-arrival times of the merged process already reveals, how the analysis explodes with each further step

$$T_{\text{out}}^{(2)} = \begin{cases} \min\left\{T_1^{(2)}, T_2^{(1)} - T_1^{(1)}\right\}, & T_1^{(1)} \leq T_2^{(1)} \\ \min\left\{T_1^{(1)} - T_2^{(1)}, T_2^{(2)}\right\}, & T_1^{(1)} > T_2^{(1)} \end{cases}, \tag{4.42}$$

$$T_{\text{out}}^{(3)} = \begin{cases} \min\left\{T_1^{(3)}, T_2^{(1)} - T_1^{(1)} - T_1^{(2)}\right\}, & T_1^{(1)} \leq T_2^{(1)}, T_1^{(2)} \leq T_2^{(1)} - T_1^{(1)} \\ \min\left\{T_1^{(2)} - \left(T_2^{(1)} - T_1^{(1)}\right), T_2^{(2)}\right\}, & T_1^{(1)} \leq T_2^{(1)}, T_1^{(2)} > T_2^{(1)} - T_1^{(1)} \\ \min\left\{T_1^{(1)} - T_2^{(1)} - T_2^{(2)}, T_2^{(3)}\right\}, & T_1^{(1)} > T_2^{(1)}, T_2^{(2)} \leq T_1^{(1)} - T_2^{(1)} \\ \min\left\{T_1^{(2)}, T_2^{(2)} - \left(T_1^{(1)} - T_2^{(1)}\right)\right\}, & T_1^{(1)} > T_2^{(1)}, T_2^{(2)} > T_1^{(1)} - T_2^{(1)} \end{cases}. \tag{4.43}$$

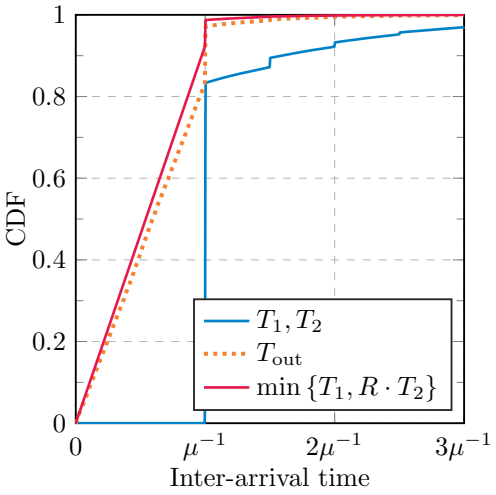


Fig. 4.9. An approximation for the distribution of two merged departure processes of two equal M/D/1 queues.

Here, the problem comes from the fact that it needs to be taken into account which process was chosen in the previous step, because this time has to be subtracted from the other one. For Poisson processes, this would not be a problem thanks to the memoryless property. Therefore, a good approximation has to be found, which constitutes a challenge for further research. If both considered processes are equal, the following approximation could serve as a starting point

$$T_{\text{out}} \approx \min\{T_1, R \cdot T_2\}, \tag{4.44}$$

with $R \sim \mathcal{U}([0,1])$. Here, the term $R \cdot T_2$ represents the RV that the inter-arrival time T_2 is already running for some time, which can be modeled by multiplying with a uniformly distributed RV. Fig. 4.9 shows numeric results for this simple approach that approximate the simulated curves. However, for more general cases with more than two inputs that are not identically distributed, more research effort has to be spent.

4.6 Modeling versus Simulation

As in the previous chapter, the modeling is put into perspective by comparing its complexity and computation time with respective simulations. Again, comparison is not trivial, due to the same aspects mentioned as in Section 3.4. In particular, no analytical solutions are available to assess accuracy. The results are described in the subsequent sections and summarized in Table 4.1.

4.6.1 Simulation

In contrast to Section 3.4, a discrete-event simulation (DES) is used, rather than a TTI-based. The DES chooses the intervals between consecutive events, i. e., arrivals, departures, and transitions in the network, which all change the current conditions. This approach is more efficient than simulating each TTI, because multiple TTIs with the same conditions can be treated at once. The number of events scales linearly with the number of packets N_F and the path length κ that represents the deepness of the network. As for each event all queues may need an update, the complexity of the DES also scales with the number of queues M. After the simulation has finished, a kernel estimation is necessary to obtain a PDF or CDF with complexity $N_\mathrm{F} N_\mathrm{G}$. The effort to calculate the percentiles from the CDF is negligible.

4.6.2 Model

On the other hand, the model complexity is governed by Algorithm 4.1 to obtain the waiting time distribution at each node and by the evaluation of Eq. (2.6), i. e., summing up the latency along a path. In case analytical expressions with negligible complexity are available, e. g., for $\mathrm{M/D}/1$ systems, Algorithm 4.1 can be skipped. The complexity of the algorithm depends on the load-dependent number of iterations N_iter (cf. Fig. 4.5b). In each iteration, a convolution of two PDFs is the dominant operation, which scales with the grid resolution as $N_\mathrm{G} \log N_\mathrm{G}$, since it can exploit an FFT. Of course, the algorithm has to be performed for each of the M queues. In some scenarios, which contain equivalent queues with the same arrival and service processes, this part can be optimized. Furthermore, the latency distributions of each node along a path of length κ have to be added, which results in another convolution to obtain the overall PDF or CDF. With the CDF, arbitrary percentiles are provided as well.

4.6.3 Low Complexity Upper Bound

Eq. (4.21) provides a low complexity formula for an upper performance bound for the CDF in an arbitrary $\mathrm{GI/GI}/1$ queue, and thus provides a fast way to obtain performance bounds for percentiles, which are the most important measure for URLLC. The only part that may require significant effort is determining the parameter θ_0 in Eq. (4.19). A binary search can find the supremum with logarithmic effort, where each evaluation of Eq. (4.20) involves a numerical

Tab. 4.1. Computational complexity and effort for queuing network models. Comparison of simulation and model.

	Simulation		Model		Upper Bound	
	$\mathscr{C}$ ($\mathcal{O}(\cdot)$)	T_{cp} [s][a]	$\mathscr{C}$ ($\mathcal{O}(\cdot)$)	T_{cp} [s][a]	T_{cp} [s][a]	$\mathscr{C}$ ($\mathcal{O}(\cdot)$)
Sim.	$N_{\mathrm{F}}\kappa M$	48.48	–	–	–	–
Algo. 4.1 [b]	–	–	$MN_{\mathrm{iter}}N_{\mathrm{G}}\log N_{\mathrm{G}}$	1.07	–	–
PDF, CDF	$N_{\mathrm{F}}N_{\mathrm{G}}$	6.36	$(\kappa-1)N_{\mathrm{G}}\log N_{\mathrm{G}}$	0.00	–	–
Percentile [b]	–	–	–	–	$\kappa N_{\mathrm{G}}\log N_{\mathrm{G}}$	0.05
Total	$N_{\mathrm{F}}(N_{\mathrm{G}}+\kappa M)$	54.84	$(MN_{\mathrm{iter}}+\kappa)N_{\mathrm{G}}\log N_{\mathrm{G}}$	1.07	$\kappa N_{\mathrm{G}}\log N_{\mathrm{G}}$	0.05

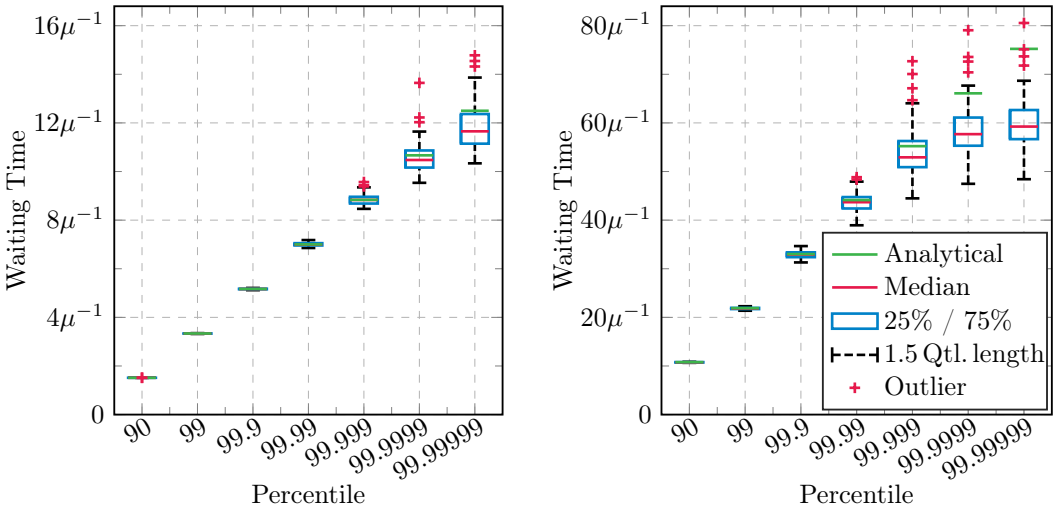
[a] The computation times are averaged over 100 runs performed on an Intel® Xeon® CPU E5-2697 v4, 2.30GHz for a single $\mathrm{M/GI/1}$ queue to have comparable results, i. e., $\kappa = M = 1$.

[b] The complexity is provided for the worst case, when no analytical results are available and thus, the terms have to be evaluated numerically.

integration of linear complexity in the grid number N_{G}. Since this operation has to be performed for each of the queues along a path, the complexity also scales with the path length κ. Again, if an analytical expression is available, the effort can be reduced to an negligible time.

4.6.4 Comparison

For a fair comparison, the computation time was measured on a single $\mathrm{M/D/1}$ system, because analytical results are available for a benchmark in this case. This reduces the convolution along a path for the model evaluation to zero, but this is the minor part of the computation anyways and would affect the simulation as well. The results in Table 4.1 were created with $N_{\mathrm{F}} = 10^7$ packets and a grid number of $N_{\mathrm{G}} = 6001$. It can be observed in Figs. 4.5c–d, which were created with the same settings, that the model is much more accurate at the distribution tail for this configuration. This behavior is underlined in Fig. 4.10 for medium (a) and high load (b). The box plots show how the results for extremely high percentiles vary. Therefore, the simulation of $N_{\mathrm{F}} = 10^7$ packets in an $\mathrm{M/D/1}$ queue was repeated 100 times with different seeds. In each simulation, different percentiles $r \in \{90, 99, 99.9, 99.99, 99.999, 99.9999, 99.99999\}\,\%$ of the waiting time were measured to study how accurate each percentile can be obtained. Box plots for the measurement of each percentile are depicted in Fig. 4.10 and it can be observed that for high network load, the variation of the simulation results increases and it becomes less likely to hit the analytical value of the high



(a) Medium load $\rho = 0.5$ (b) High load $\rho = 0.9$

Fig. 4.10. Accuracy of simulation results to determine extremely high percentiles. The box plots show how percentiles vary in 100 runs with $N_\mathrm{F} = 10^7$ simulated packets each. A green line indicates the true (analytical) values from Eq. (4.23).

percentiles. The proposed algorithm, which does not suffer from this problems, still achieves a much lower computation time. The model is still valid for even higher percentiles, which would require several orders of magnitude more of simulation time. Finally, the low complexity upper bound, which is quite tight (cf. Figs. 4.5c–d), enables a 20 times faster calculation of an upper bound for the percentiles than the model. However, this comes at the cost of accuracy and knowledge about the entire distribution.

It can be summarized that the model outperforms simulations not only in terms of the computation time by a factor of 50 but also in the achieved accuracy at the distribution tail. As shown in Fig. 4.10, the values of high percentiles obtained from simulation constitute a large spread around the analytical value. Consequently, simulations are not appropriate for the tail evaluation as required in the URLLC domain. Instead, mathematical models like the ones provided in this thesis can fill the gap.

4.7 Summary and Conclusion

This chapter focuses on the architecture behind the wireless access with the aim of providing a framework for E2E latency evaluation. One of the key contributions of this thesis is constituted by the development of a numerical method that delivers the latency distribution of arbitrary $\mathrm{GI}/\mathrm{GI}/1$ systems, i. e., systems

with general i.i.d. arrivals and service processes. This enables performance evaluation of general network nodes as long as the governing random processes are known. In particular, by generating the entire distribution, all percentile values are inherently provided, which are required for URLLC applications, but hard or even infeasible to be obtained by simulation. Future work should aim for percentiles as well. Further studies may extend the method to even more general systems, such as integrating admission control ($\mathrm{GI/GI/1}/K$), multiple servers ($\mathrm{GI/GI/c}$), other queuing disciplines, or combinations of those.

In case speed is more important than accuracy, an upper performance bound for the percentiles of $\mathrm{GI/GI/1}$ nodes has been provided as well, which can be relevant for live network optimization, where results are immediately required. As an upper performance bound, required guarantees will not be hurt.

In regard of 5G use cases, heterogeneous traffic from multiple applications is expected to be present simultaneously. Even though logically separated through network slicing, different classes still use the same physical infrastructure and resources, which need to be allocated and, thus, require efficient scheduling. For this purpose, four different schedulers were integrated into the model. Therein, good approximations for results from simulations could be determined. Since each of the schedulers pursues different targets, the modeling results may help in choosing the appropriate one for the particular aim at hand. Even though the analysis in this thesis is restricted to two classes, approaches were provided for the model extension to three or more classes. Furthermore, the modeling may be a starting point for studying further, potentially more sophisticated scheduling policies.

Again, the modeling was put into perspective, by comparing the computational effort and complexity and thereby verifying that the initial aim of having faster evaluation methods could be achieved. Therein, the model and the upper bound turned out to be 50 and 1000 times faster than simulation, respectively, while providing better accuracy for high percentiles.

Finally, the chapter presents remaining open questions and research challenges for generalizing the E2E latency evaluation framework. Therein, approaches were sketched that could build the foundation of future research.

5

Latency Improvement in a Realistic Scenario

After studying traffic originating from a whole set of applications and capturing the radio interface as well as the RAN in mathematical models, the gained insights should now be used to discuss concepts to improve network performance of a more realistic scenario. Even though the topology and parameters are tailored to a specific scenario for this application, the settings could be easily changed to investigate other scenarios as well.

The approaches for performance improvements explained in this chapter could serve as a part of the *self-optimization* aspect of SONs. The SON concept [NGM08b] has the aim of increasing efficiency, especially when networks of high complexity are considered. When network complexity grows, management and configuration becomes cumbersome or even intractable for manual adjustment by humans. Thus, the idea of SON is that the network chooses its parameters autonomously according to the current conditions, which typically change dynamically over time. This way, the network is expected to provide better performance or to run more efficiently while less human intervention is necessary and, thus, OPEX can be saved. SON was included into the LTE standard in [3GP11c].

According to [Ber+14; Ber15], SON can be classified into online and offline SON. In online SON approaches, network KPIs are permanently measured and fed into a SON algorithm. Based on the measurements, the SON algorithm determines new network parameters and reconfigures the network accordingly, which completes the loop. In contrast, offline SON does not apply the resulting parameters directly to the network but to a mathematical model or a simulation of the network. This way, the optimization, which usually involves several iterations, runs in an offline feedback loop, without affecting the running network. When the offline optimization method has found parameters that improve the estimated performance by the model, the resulting parameters will be applied to

the real network. Since offline SON needs models for network performance estimation, it can benefit from the approaches introduced in the previous chapters of this thesis.

Both approaches have advantages and disadvantages. Online SON usually needs less detailed measurements, because there is no model that has to be parameterized. However, the choice of possible optimization algorithms is limited, since the parameters are applied to the real network and therefore misconfiguration should be avoided. Thus, stable algorithms are required, which usually change the configuration only slowly. Furthermore, the optimization process may take a long time, because the algorithm has to wait for new measurements between consecutive iterations. In contrast, offline SON can use arbitrary algorithms that can have bad parameter choices in intermediate steps without harming the actual network. Moreover, the time between iterations is only limited by the evaluation time of the model. Since parameters are tested with the model in advance, misconfiguration of the network is expected to be less likely. However, offline SON depends on accurate modeling, which may require more detailed measurements of the network as an input to the model.

Similar to a Kalman filter [Kal60], both offline and online SON could also be combined by letting the model predict the network performance and taking live measurements to correct the prediction. However, this approach is left for future studies.

First, the scenario of interest along with the formulation of the optimization problem is described. Afterwards, optimization approaches are explained and evaluated.

5.1 Problem Formulation

This section explains the considered scenario. The targets of the performance improvement can be mathematically formulated as an optimization problem. Furthermore, the available parameters are identified.

5.1.1 Scenario

The concrete cellular scenario for which the performance improvement will be conducted is illustrated in Fig. 5.1. It consists of 21 BS sites, where each one is equipped with sectorized antennas spanning three cells per BS. For the

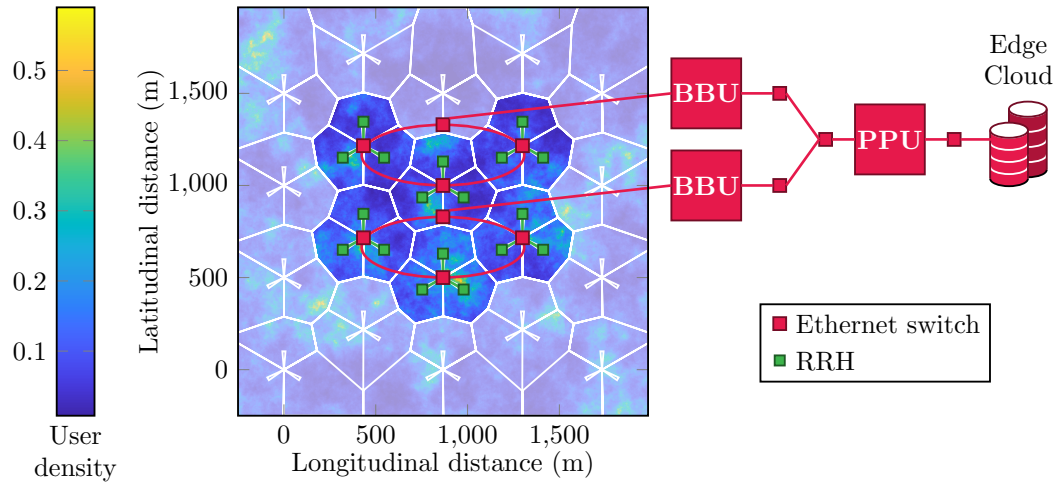


Fig. 5.1. Specific cellular scenario for the performance improvement with an exemplary realization of the user density. Two ring topologies connect the BS sites to the BBUs, a PPU, and an edge cloud server. The surrounding cells (shaded) are not included in the queuing network analysis but create realistic interference conditions for the considered cells in the center.

performance evaluation, six sites (i. e., 18 sectors) in the center of the scenario are considered. The purpose of the surrounding sites is to create reasonable interference conditions. A realistic user distribution is obtained from the spatial traffic modeling introduced in Section 2.1.2. The wireless access is modeled the same way as in Section 3.1, i. e., the formulas in Eqns. (3.4)–(3.6) are taken for the path loss, the SINR and the achievable rate, respectively. Other simulation parameters are aligned with the ones in Chapter 3 as well.

The six considered BS sites are connected in two ring topologies, comprising three sites each. The ring topologies offer a highly reliable connection. If one of the links between the antenna sites breaks, there is still an alternative route. By using the Ethernet switch model described in Section 4.1.2, infinite loops in the routing can be avoided, by removing the respective connections. For instance, a packet is not allowed to be routed back to its origin within a switch.

The rings are each connected to one of two BBUs (base band units, cf. Section 4.1), which are in turn linked to a common PPU (packet processing unit, cf. Section 4.1). Finally, the PPU is connected to a cloud server. In a real mobile network, this architecture would comprise much more rings, BBUs, and PPUs. This may even lead to more flexibility through a full mesh network between the BBUs and PPUs. However, for the analysis conducted here, the scenario is restricted to this level of complexity.

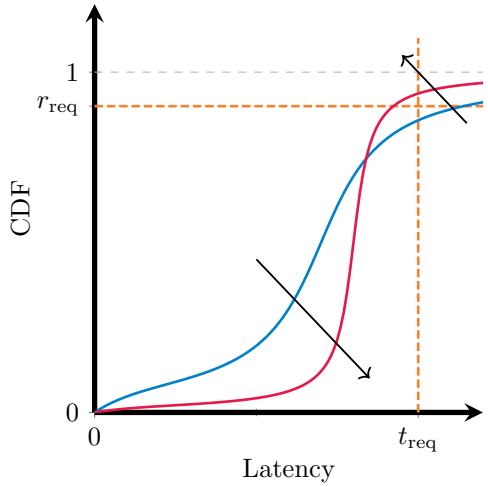


Fig. 5.2. Sketch of CDF reshaping with respect to (O1). For URLLC, the CDF should lie above the point defined by the required latency t_{req} for the given percentile r_{req}.

Two different traffic classes are considered with WFQ scheduling, which was chosen due to the higher flexibility through the scheduling weights compared to priority queuing. Again, class $\mathfrak{c} = 1$ is regarded as the application with higher priority. Both classes are assumed to have Poisson arrivals and fixed packet sizes. Due to the mentioned open questions in Section 4.5, the subsequent evaluations will be based on simulations.

5.1.2 Optimization Problem

With regard to URLLC applications, the following two approaches are identified within this thesis for a network performance improvement:

(O1) minimize the latency of a certain percentile r_{req} for the high priority class, or

(O2) take a given latency requirement t_{req} for a certain percentile r_{req} of the high priority class as a constraint and improve another KPI, e. g., the latency of another slice.

As mentioned earlier (cf. Section 2.2.1), for URLLC it is of great importance to consider the high percentiles of the latency distributions for the performance improvement rather than, e. g., the mean latency. For this purpose it is also legitimate to degrade the performance of the lower part of the distribution, because there is no benefit in achieving values far below the given requirement. In other words, the aim is to reshape the latency distribution as sketched in

Fig. 5.2. Given a required latency t_{req} for a high percentile r_{req}, a vertical and a horizontal line in the diagram are defined, respectively. With this visualization, the only important requirement is that the CDF crosses the latency threshold above the line defined by the percentile to meet the URLLC demands. To achieve this, the left part of the curve can be arbitrarily deformed. An example on how this can be done is already given in the traffic offloading example in Fig. 4.3.

With this in mind, the two performance targets (O1) and (O2) can be seen as two steps. The target (O1) can be pursued to bring the system into a valid state first, i. e., to find a configuration that allows the network to meet the latency requirements. Once being in this state, the second target (O2) is reasonable to improve the network performance, without harming the given latency constraints.

Let $\mathcal{P}$ denote the set of parameters that can be changed in order to improve the network performance. In the scenario at hand, the following parameters for performance improvement have been identified ($i, j \in \mathcal{M}$):

- the routing matrix $\boldsymbol{P}^{(\mathfrak{c})} = \left(p_{ij}^{(\mathfrak{c})}\right)$,
- admission control parameters $K_i^{(\mathfrak{c})}$, and
- the scheduling weights $w_i^{(\mathfrak{c})}$.

As admission control leads to packet dropping, which is expected to be counterproductive for URLLC, because only very low outage can be tolerated, this parameter is not considered. Thus, the parameter set is $\mathcal{P} = \left\{p_{ij}^{(\mathfrak{c})}, w_i^{(\mathfrak{c})} \mid i, j \in \mathcal{M}\right\}$. The only constraint on the parameter space is that the probabilities and weights have to sum up to one, respectively.

The target (O1) refers to the following optimization problem. Maximize the percentile for which the latency constraint t_{req} is met, i. e.,

$$\max_{\mathcal{P}} F_J^{(1)}\left(t_{\mathrm{req}} \mid \mathcal{P}\right) \tag{5.1}$$

with respect to the probability constraints

$$\boldsymbol{P}^{(\mathfrak{c})}\mathbf{1} = \mathbf{1}, \qquad w_i^{(\mathfrak{c})}\mathbf{1} = 1 \qquad \forall i \in \mathcal{M}, \mathfrak{c} \in \mathcal{C}. \tag{5.2}$$

For the other target (O2), the target function from Eq. (5.1) becomes a constraint and another function g that represents the KPI of interest will be improved.

$$\max_{\mathcal{P}} g\left(\mathcal{P}\right) \tag{5.3}$$

with respect to

$$F_J^{(1)}\left(t_{\mathrm{req}} \mid \mathcal{P}\right) \geq r_{\mathrm{req}} \qquad \text{and Eq. (5.2).} \tag{5.4}$$

Here, Eq. (5.3) can also refer to minimize a KPI, e. g., latency. To formulate it as a maximization problem, the target function can be multiplied by minus one.

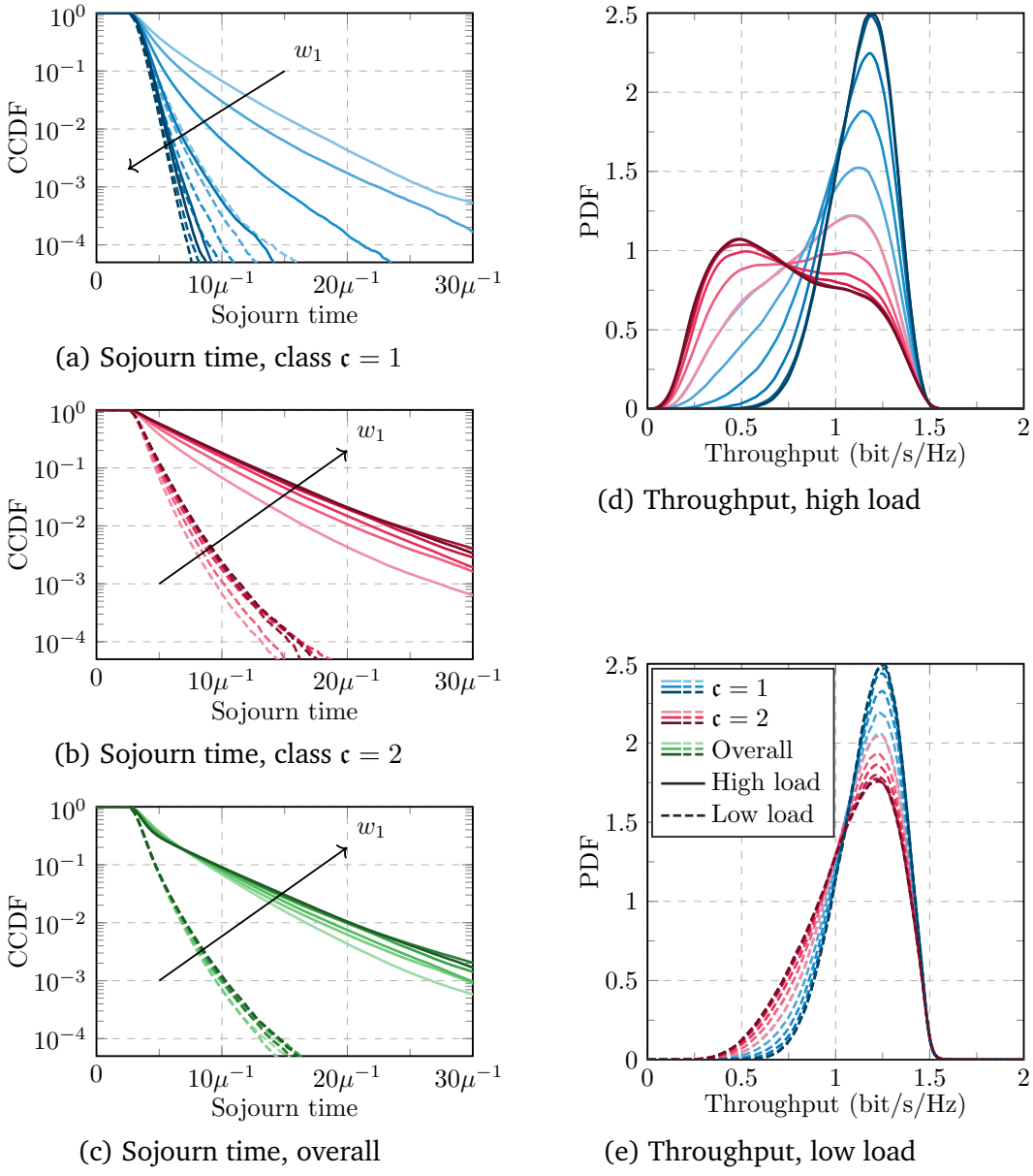
5.2 Approach for Performance Improvement

As the identified optimization problems are analytically not tractable, i. e., there is no known analytical solution, no optimization is conducted here in the mathematical sense. Instead, the aim is to improve the network performance as far as possible. Therefore, the impact of the scheduling weights on the performance is investigated first. Afterwards, an adaptive scheduling is introduced to decrease the latency.

5.2.1 Impact of Scheduling Weights

Fig. 5.3 shows the complementary cumulative distribution function (CCDF) of the E2E latency for both classes for different network loads and different configurations of the WFQ scheduler. The results are obtained from simulations. As it can be seen in Fig. 5.3a, the weight of class $\mathfrak{c} = 1$ is a good means to control the latency percentiles of the high priority class. With higher prioritization, the high load curves can be even pushed towards the ones of low load scenarios. Obviously, $w_1 = 1$ minimizes the latency for all percentiles and, thus, solves the optimization problem in Eq. (5.1) for fixed routing probabilities. This is due to the fact that $w_1 = 1$ fully prioritizes class $\mathfrak{c} = 1$ over class $\mathfrak{c} = 2$, which leads to the lowest latency, because there are no constraints that hinder choosing the most extreme value.

However, prioritizing class $\mathfrak{c} = 1$ adversely impacts the performance of class $\mathfrak{c} = 2$ (see Fig 5.3b) and the overall high percentile performance (Fig 5.3c). Even if the latency of class $\mathfrak{c} = 2$ is not of great importance, one can also see the negative impact on throughput in Figs. 5.3d–e, especially for the high load scenario.



(a) Sojourn time, class $\mathfrak{c} = 1$

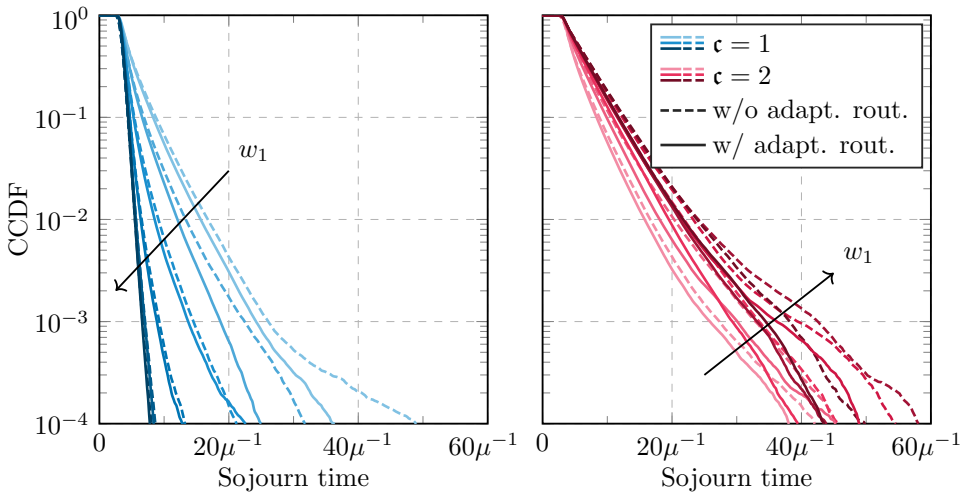
(b) Sojourn time, class $\mathfrak{c} = 2$

(c) Sojourn time, overall

(d) Throughput, high load

(e) Throughput, low load

Fig. 5.3. Impact of scheduling weights for different loads on the sojourn time of (a) class 1, (b) class 2, and (c) the overall traffic, as well as on the throughput of both classes for (d) high ($\rho = 0.8$) and (e) low load ($\rho = 0.3$), respectively.



(a) Class $\mathfrak{c} = 1$ (b) Class $\mathfrak{c} = 2$

Fig. 5.4. Impact of adaptive routing on the latency of both traffic classes. Latency distribution of (a) class $\mathfrak{c} = 1$ and (b) of class $\mathfrak{c} = 2$.

Thus, the weight w_1 should only be increased as far as necessary to meet the application requirements.

As there are low percentiles with good performance that could be degraded without harming URLLC requirements one could think of dynamically adjusting the scheduler weights according to the current performance, i. e., increasing w_1, when performance is too bad or increasing it, when it is very good. This could be realized in a centralized or in a distributed way. However, it turned out that those strategies do not lead to significant performance gains for any of the classes. This can be explained as follows. Situations that allow relaxing the weight of class $\mathfrak{c} = 1$ come along with a low load of high priority traffic. Accordingly, lowering w_1 has only small impact in such situations as there is not much traffic the other class has to compete with.

5.2.2 Adaptive Routing

As the second identified option to improve performance, routing is considered in this section. Here, an intuitive approach is to adapt the routing to the load at neighboring nodes, which can be done in a distributed manner. At each node, the routing decision will be made upon the load at the possible nodes where the traffic can be forwarded to, i. e., the neighbor with the smallest load will be chosen as the target. Herein, the load comprises the traffic of both classes jointly.

Tab. 5.1. Sojourn time reduction through adaptive routing for both traffic classes in different percentiles of interest.

w_1	$r_1 = 99.9, \mathfrak{c} = 1$			$r_2 = 99, \mathfrak{c} = 2$		
	w/o adapt.	w/ adapt.	reduction (%)	w/o adapt.	w/ adapt.	reduction (%)
0.50	26.08	24.19	-7.24	16.63	15.77	-5.17
0.55	24.71	24.51	-0.82	18.45	16.85	-8.65
0.60	22.45	18.70	-16.67	20.32	18.08	-11.04
0.65	17.92	16.11	-10.10	21.33	19.05	-10.67
0.70	14.53	13.70	-5.72	21.85	19.50	-10.78
0.75	11.54	11.23	-2.73	23.67	21.18	-10.55
0.80	9.25	8.88	-4.03	23.27	21.49	-7.65
0.85	7.81	7.64	-2.13	23.64	21.38	-9.56
0.90	7.06	7.01	-0.71	24.06	21.19	-11.94
0.95	6.73	6.69	-0.61	23.33	21.59	-7.46
1.00	6.67	6.60	-1.02	23.86	21.71	-8.99

The results of this approach can be seen in Fig. 5.4. The subfigures depict the CCDFs of both classes for different values for the scheduling weights with and without the adapted routing, respectively. It can be observed that the performance improves for both classes, as the solid CCDFs are further left than the dashed lines. Table 5.1 summarizes the performance gains. For the high priority class a higher percentile ($r_1 = 99.9$) is considered than for the lower priority class ($r_2 = 99$), because in the high priority case, low outage probability is of importance, whereas for the other traffic a majority shall experience good performance, but outliers at the distribution tail can be tolerated. For all evaluated scheduling weights, a gain can be observed in both considered metrics. In particular, the 99$^{\text{th}}$ percentile of class $\mathfrak{c} = 2$ experiences gains in the order of $10\,\%$ in almost all of the considered cases. In current LTE networks (cf. Section 1.2) this would translate to saving around $4\,\text{ms}$ and, thus, would not be sufficient. Recent work [Tri+15] achieved $30\,\%$ latency reduction on the user plane of an LTE network through an SDN architecture. As both approaches are not contradicting, they could also be combined.

The high priority class $\mathfrak{c} = 1$ also benefits from the adaptive routing. However, it can be seen again that the performance of class $\mathfrak{c} = 1$ can be controlled primarily by the scheduling weights. Thus, w_1 can be chosen, such that the application requirement is met, before improving the performance of the other class. Interestingly, those gains could be observed, even though this strategy

might mean that traffic has to experience more hops by taking an alternative route.

5.3 Summary and Conclusion

In this chapter, offline and online approaches for SONs have been explained. In this regard, it has been concluded that offline SON may benefit from the performance evaluation models introduced in the previous chapters.

In general, latency optimization for critical communications should target in minimizing very high percentiles to ensure very low outage probability due to outdated packets. With demanding URLLC requirements, this should be taken rather as an optimization constraint than as the target to be optimized. It turned out that scheduling weights are an appropriate means to control the latency percentiles of high priority traffic. By giving the full weight to high priority traffic, latency percentiles can be minimized. However, as this may lead to a latency far below the required threshold, less strict values may be the better choice. By taking the requirement as a constraint, the latency or another KPI such as throughput of the other class can be improved instead.

As another result, load-dependent routing turned out to be an efficient means to reduce latency for all traffic classes. However, the investigated approach considered only neighboring nodes. A centralized solution that has an overview of the entire network or at least the possible paths is expected to result in better performance. Also the complexity of the studied network is low. A bigger network that enables more routing options may also have a greater potential for performance improvements.

Further potential for performance improvement is also expected by enriching the model with more technical details that can be tuned for better network performance. For instance, with the introduction of NFV, network functions will run on commercial of the shelf (COTS) hardware. This not only enables flexible deployment at different locations in the network, but also different allocation of computation resources to multiple functions. This offers a whole set of new DoFs to optimization problems. In this regard, it would be also interesting to add further contributors to latency apart from the queuing delays into the model. However, theses aspects are out of scope for this thesis and left for further studies.

6 Conclusions & Outlook

The focus of this thesis is laid on E2E latency modeling for future mobile networks. Therefore, the mathematical framework of queuing theory and queuing networks is chosen and introduced in the beginning of this thesis. Since system dynamics are strongly influenced by the traffic behavior of considered applications, foundations for spatial and temporal traffic modeling are presented as well. With these tools, the radio access in the DL under the influence of cellular dynamics is studied. Furthermore, queuing models for homogeneous and prioritized heterogeneous traffic are developed to analyze latency in general systems. Afterwards, a realistic cellular scenario is considered in which the latency performance is improved. In the following, key results are summarized and conclusions are drawn. Finally, recommendations for future work are provided.

6.1 Key Results and Conclusions

A MATLAB [Mat19] class has been implemented to handle stochastic distributions in a convenient way. It allows operations on independent RVs, which are implemented analytically, if an explicit expression is known, and numerically otherwise, to keep general applicability. The implementation may serve as a powerful tool for future research as well.

Transient models for video streaming performance in terms of the startup delay (latency) distribution and buffer starvation probability (reliability) are extended to integrate multi-cellular interference dynamics. To solve the arising PDE systems, an appropriate numerical solver has been proposed, since existing approaches suffer from numerical problems (e. g., oscillations) or oversimplified analysis. Interestingly, in the application of video streaming, it turns out that the buffer threshold is only appropriate to control the startup delay but has rather limited impact on the starvation probability, since the video will interrupt eventually, if receiving conditions are bad and it is sufficiently long. However, the framework provides alternatives with great impact, such as reducing the

video bitrate in the considered or in the neighboring cells. The models can provide valuable insights for network performance also for other use cases, if they are adjusted accordingly.

Queuing networks are proposed as a general framework for evaluating arbitrary (wireless) network topologies with respect to E2E latency. By allowing general $\mathrm{GI}/\mathrm{GI}/1$ queuing systems for modeling the single network nodes as well as different schedulers for prioritized heterogeneous traffic of multiple applications, the framework is designed to be very flexible. However, identified open questions underline the demand for further research in this area. For instance, it has to be studied how passing queues, splitting, and merging changes the traffic distributions. With respect to URLLC it is identified that performance modeling should aim for obtaining high percentiles or even entire distribution functions (PDF or CDF) as a performance metric, since average values do not provide sufficient information about critical outliers. Therein, the strong impact of queuing effects on latency is confirmed.

Finally, two optimization approaches with respect to URLLC are identified: (1) minimizing the latency and (2) taking the latency requirements for a given percentile as a constraint to optimize another KPI, e. g., throughput of low priority traffic. Both approaches refer to a desired reshaping of latency distributions. Based on an exemplary multi-cellular scenario with a realistic user distribution, the impact of scheduling weights and routing in a RAN is studied. The scheduling weights are identified as a powerful tool to control the latency of high priority traffic (at the cost of other applications). However, an adaptive adjustment of these weights can only provide minor benefits. In contrast, adaptive routing turns out to provide performance gains for both considered traffic classes.

6.2 Recommendations for Future Work

Throughout the thesis, opportunities for further studies are mentioned where suitable. These are summarized as follows.

- The development of the distribution class can be continued, since its implementation was driven by the functionalities required for the thesis so far. This includes adding further standard distributions or operations. Another possible extension would be constituted by allowing different numerical grids for each instance of the class.

- The results of the theoretical framework can be validated with testbed or live network measurements to complement the validation through simulations. Therein, further technical details of actual implementations could be integrated into the model, which is already prepared to accommodate technical delays.
- With this enrichment of the model by technical details, further optimization potential is expected which allows for complementary research.
- The presented approach for UL flow-level modeling could be further pursued, which requires finding appropriate solvers for the involved non-linear system as well as the implementation of a simulation (or using an existing simulator) to verify the results.
- The developed algorithm to determine the waiting time distribution of $\mathrm{GI/GI/1}$ queues could be extended to the even more general systems, such as the $\mathrm{GI/GI/c}$ queue with multiple servers or other queuing disciplines than FIFO.
- For the heterogeneous traffic case, different schedulers can be added into the model to investigate their performance. Moreover, the presented scheduling models could be extended to more than two simultaneous traffic classes, for which modeling approaches are provided in this thesis.
- The open questions for the queuing network traffic modeling as discussed in Section 4.5 need to be addressed. It has to be studied how passing queues as well as splitting and merging the traffic changes the traffic distributions. There approaches were already sketched to serve as a starting point.
- Although, it has been empirically shown that the Kleinrock independence approximation holds for dense networks, further mathematical tools can be developed to handle the correlated sojourn times of queues along a path.

A Appendix

A.1 Simulation Parameters

Tab. A.1. System Parameters (if not explicitly stated otherwise).

Description	Symbol	Value
Network Parameters (aligned with [Kle+16; 3GP10])		
Maximum transmit power	p^{tx}	$49\,\mathrm{dBm}$
Carrier frequency	f_c	$2\,\mathrm{GHz}$
Bandwidth	B_{BW}	$20\,\mathrm{MHz}$
Bandwidth efficiency	e_{BW}	0.63
SINR efficiency	e_{SINR}	0.4
Path loss [3GP10]	$l_{\mathrm{path}}(d)$	$128.1\,\mathrm{dB} + 37.6 \log_{10}\left(\frac{d}{\mathrm{km}}\right)\mathrm{dB}$
Thermal noise	N_0	$-174\,\mathrm{dBm\,Hz^{-1}}$
Fast fading margin [FRF09]	G_{FF}	$-2\,\mathrm{dB}$
Antenna diversity gain [FRF09]	G_{div}	$-3\,\mathrm{dB}$
BS noise figure [3GP18b]	G_{BS}	$-5\,\mathrm{dB}$
UE noise figure [3GP18b]	G_{UE}	$-9\,\mathrm{dB}$
BS height [3GP10]	h_{BS}	$32\,\mathrm{m}$
UE height [3GP10]	h_{UE}	$1.5\,\mathrm{m}$
Video Traffic		
Temporal user distribution	A	Poisson
Spatial user distribution	$f_{\boldsymbol{u},i}$	$\mathcal{U}$
Mean system arrival rate	λ	$0.0375\,\mathrm{s^{-1}}$
Video duration	T_{video}	$\sim$ Exp
Mean video duration	$\bar{T}_{\mathrm{video}}$	$240\,\mathrm{s}$
Video bitrate	R_{CBR}	$2\,\mathrm{Mbps}$
Traffic demand (center)	T_{center}	$18\,\mathrm{Mbps}$
Traffic demand (neighbors)	T_{neighbor}	$18\,\mathrm{Mbps}$

A.2 Further Tests of the GI/GI/1 Algorithm

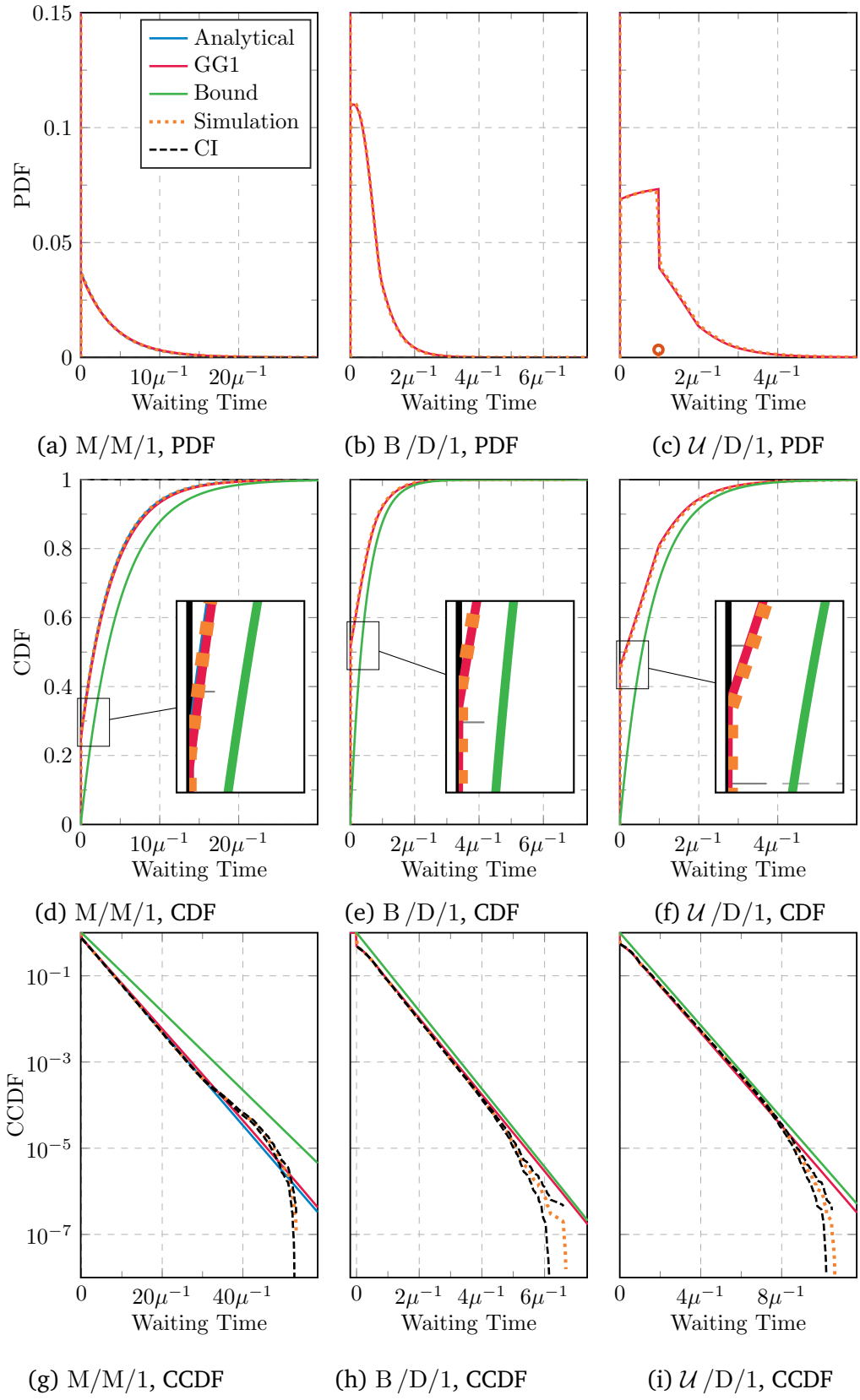


Fig. A.1. Algorithm 4.1 tested with other queuing models. Analytical results only exist for M/M/1. The impulses at zero in the PDFs are not visible due to the scaling, but one can equivalently check at which point the CDFs intersect the y-axis.

A.3 Proof of Proposition 4.1

Proof of Prop. 4.1. In this proof, the convergence in distribution is shown, i. e., there exists an RV W for which the following holds

$$\lim_{i\to\infty} F_{W_n}(t) = F_W(t), \qquad \forall t \in \mathbb{R}. \tag{A.1}$$

As a first step, the monotonicity $F_{W_n}(t)$ with respect to n is demonstrated, by showing that

$$F_{W_{n+1}}(t) \leq F_{W_n}(t), \qquad \forall t \in \mathbb{R} \tag{A.2}$$

holds for all $n \in \mathbb{N}$.

By construction, $F_{W_n}(t) = 0$ for all $t < 0, n \in \mathbb{N}$, such that Eq. (A.2) holds in this case. Now, the case $t \geq 0$ is considered. For $n = 0$, it holds that $W_0 = 0$ and, thus, $F_{W_0}(t) = \chi_{[0,\infty)}(t)$. Hence, Eq. (A.2) holds for $n = 0$. For $n > 0$ the construction of the RVs is considered as follows

$$W_0 = 0 \tag{A.3}$$
$$W_1 = \max\{W_0 + U_0, 0\} = \max\{U_0, 0\} \tag{A.4}$$
$$W_2 = \max\{W_1 + U_1, 0\} = \max\{\max\{U_0, 0\} + U_1, 0\} \tag{A.5}$$
$$= \max\{\max\{U_0 + U_1, U_1\}, 0\} \tag{A.6}$$
$$= \max\{U_0 + U_1, U_1, 0\} \tag{A.7}$$
$$W_3 = \max\{W_2 + U_2, 0\} = \max\{\max\{U_0 + U_1, U_1, 0\} + U_2, 0\} \tag{A.8}$$
$$= \max\{\max\{U_0 + U_1 + U_2, U_1 + U_2, U_2\}, 0\} \tag{A.9}$$
$$= \max\{U_0 + U_1 + U_2, U_1 + U_2, U_2, 0\} \tag{A.10}$$
$$\vdots$$
$$W_n = \max\left(\left\{\sum_{l=k}^{n-1} U_l \middle| k = 0, \ldots, n-1\right\} \cup \{0\}\right) \tag{A.11}$$

Herein, always the result from the previous line is inserted and the last line (A.11) can be easily proven by induction.

With this, the relation (A.2) is shown inductively by exploiting the relation:

$$\max\{U_0 + U_1, U_1, 0\} \geq \max\{U_1, 0\} \tag{A.12}$$
$$\mathbb{P}\left[\max\{U_0 + U_1, U_1, 0\} > t\right] \geq \mathbb{P}\left[\max\{U_1, 0\} > t\right]. \tag{A.13}$$

Accordingly, the inequality (A.2) can be derived for $n = 2$ as follows

$$
\begin{aligned}
F_{W_2}(t) &= \mathbb{P}\left[W_2 \leq t\right] && \text{(A.14)}\\
&= 1 - \mathbb{P}\left[W_2 > t\right] && \text{(A.15)}\\
&= 1 - \mathbb{P}\left[\max\{U_0 + U_1, U_1, 0\} > t\right] && \text{(A.16)}\\
&\leq 1 - \mathbb{P}\left[\max\{U_1, 0\} > t\right] && \text{(A.17)}\\
&= 1 - \mathbb{P}\left[\max\{U_0, 0\} > t\right] && \text{(A.18)}\\
&= 1 - \mathbb{P}\left[W_1 > t\right] && \text{(A.19)}\\
&= \mathbb{P}\left[W_1 \leq t\right] = F_{W_1}(t). && \text{(A.20)}
\end{aligned}
$$

Herein, (A.13) was used to obtain line (A.17). Furthermore, line (A.18) holds, since U_0 and U_1 are i.i.d.. Generalizing this step for an arbitrary $n \in \mathbb{N}$ completes the induction. This can be done with the following relation:

$$
\begin{aligned}
W_n &= \max\left(\left\{\sum_{l=k}^{n-1} U_l \middle| k = 0, \ldots, n-1\right\} \cup \{0\}\right) && \text{(A.21)}\\
&\geq \max\left(\left\{\sum_{l=k}^{n-1} U_l \middle| k = 1, \ldots, n-1\right\} \cup \{0\}\right) && \text{(A.22)}\\
&\overset{d}{=} \max\left(\left\{\sum_{l=k}^{n-2} U_l \middle| k = 0, \ldots, n-2\right\} \cup \{0\}\right) && \text{(A.23)}\\
&= W_{n-1}, && \text{(A.24)}
\end{aligned}
$$

where $\overset{d}{=}$ means equality in distribution, and performing the same steps as for the case $n = 2$. Thus, Eq. (A.2) holds for all $n \in \mathbb{N}$.

As a second step, the result derived from [Kin64] in Eq. (4.21) ensures that F_{W_n} is lower-bounded (as a CDF it is also lower-bounded by zero). Thus, $F_{W_n}(t)$ is monotonically decreasing with respect to n and lower-bounded for any $t \in \mathbb{R}$. Consequently, it must converge to a limit $F(t)$. This limiting function defines the CDF for the desired RV W. □

A.4 Linear Transport Equations

This section provides a primer on linear transport equations and approaches to solve them analytically if possible. It also shows, what might have gone wrong in the derivation of [Xu+16] (cf. Section A.4.2) and what happens if the explicit solution is inserted into the original PDE system (cf. Section A.4.3).

A.4.1 One-dimensional Case

A one dimensional linear transport equation has the form

$$\frac{\partial}{\partial t}U(t,q) + v\frac{\partial}{\partial q}U(t,q) = f(U(t,q)) \tag{A.25}$$

or shortly,

$$U_t + vU_q = f(U) \tag{A.26}$$

with $U : \mathbb{R}^2 \to \mathbb{R}, v \in \mathbb{R}, f : \mathbb{R} \to \mathbb{R}$. This can be solved by considering an auxiliary function $\xi(s)$ defined by

$$\xi(s) = U(t_0 + s, q_0 + vs). \tag{A.27}$$

The function is chosen in a way that its derivative equals the left-hand side (LHS) of Eq. (A.26)

$$\begin{aligned}\xi'(s) = \frac{\mathrm{d}}{\mathrm{d}s}\xi(s) &= \frac{\mathrm{d}}{\mathrm{d}s}U(t_0 + s, q_0 + vs) && \text{(A.28)}\\ &= U_t(t_0 + s, q_0 + vs) + vU_q(t_0 + s, q_0 + vs) && \text{(A.29)}\\ &\overset{\text{(A.26)}}{=} f(U(t_0 + s, q_0 + vs)) = f(\xi(s)). && \text{(A.30)}\end{aligned}$$

Along the so called *characteristic* $x(s) = (t_0 + s, q_0 + vs)$ (see Fig. A.2), the PDE (A.26) reduces to the ODE

$$\xi'(s) = f(\xi(s)) \tag{A.31}$$

with initial conditions

$$\xi(0) = U(t_0, q_0). \tag{A.32}$$

Thus, for a homogeneous PDE (i. e., $f(U) = 0$) the solution of Eq. (A.26) is constant along the characteristics. In this case the initial values are only *transported* along the characteristics.

Otherwise the ODE in Eq. (A.31) determines how the initial values change along the characteristics. For instance, in the special case $f(U) = -\lambda U$ the solution would be

$$\xi(s) = e^{-\lambda s}U(t_0, q_0). \tag{A.33}$$

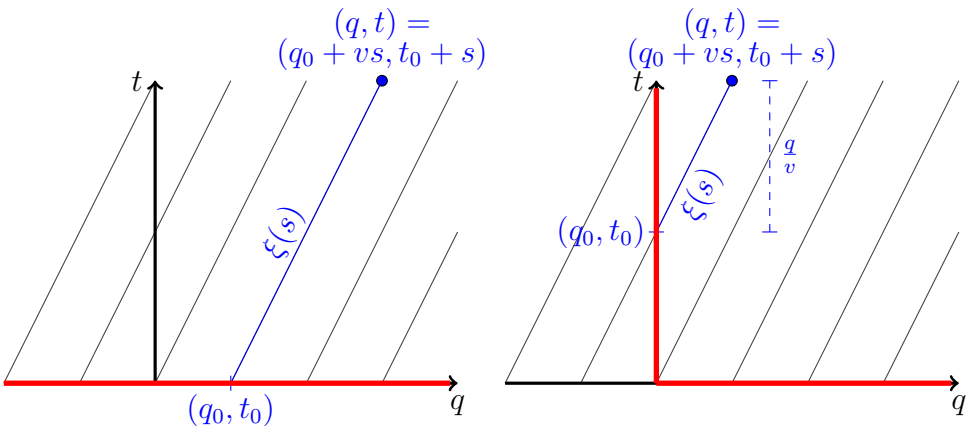


Fig. A.2. Characteristics of the PDE (A.26) for $v > 0$ with given (a) initial conditions or (b) initial and boundary conditions as indicated in red.

Initial conditions

There are different ways to specify initial or boundary conditions, which lead to different conditions for $\xi(s)$. Here, two cases are considered as illustrated in Fig. A.2.

(a) Initial conditions for U at $t = 0$ are given:

$$U(0, q) = U_0(q) \qquad \forall q \in \mathbb{R}. \tag{A.34}$$

For a given point (t, q), the respective curve parameter s and initial coordinate q_0 (which "chooses" the appropriate characteristic) can be determined as follows

$$(t, q) = (t_0 + s, q_0 + vs) \tag{A.35a}$$
$$t_0 = 0 \Rightarrow s = t \tag{A.35b}$$
$$\Rightarrow q_0 = q - vt \tag{A.35c}$$

Considering the special case of $f(U) = -\lambda U$ again, the solution can be formulated as

$$\xi(s) = e^{-\lambda s} U(0, q_0) = e^{-\lambda s} U_0(q_0), \tag{A.36a}$$
$$U(t, q) = \xi(s = t) = e^{-\lambda t} U_0(q - vt). \tag{A.36b}$$

(b) Initial conditions for $q > 0$ and boundary conditions for $q = 0$, i. e., conditions on the positive axes (requires $v > 0$ to have a well-defined problem). This is the interesting case for this thesis, since the initial conditions in Eq. (3.18) for the streaming PDE system are formulated this way.

$$U(0, q) = U_0(q) \qquad \forall q > 0 \tag{A.37a}$$
$$U(t, 0) = U_1(t) \qquad \forall t > 0 \tag{A.37b}$$

Now, two different cases have to be considered, depending on which axis intersects the respective characteristic first. (cf. Fig. A.2)

- $q - vt > 0$: Intersection with q-axis $\Rightarrow t_0 = 0$, analog to (A.35), (A.36)
- $q - vt < 0$: Intersection with t-axis $\Rightarrow q_0 = 0$

$$q - vt + vt_0 = 0 \qquad \Rightarrow t_0 = t - \frac{q}{v}, \qquad s = t - t_0 \tag{A.38}$$

Considering the special case $f(U) = \lambda U$ again yields

$$\xi(s) = e^{-\lambda s} U(t_0, 0) \qquad = e^{-\lambda s} U_1(t_0), \tag{A.39}$$
$$U(t, q) = \xi(s = t - t_0)) \qquad = e^{-\lambda (t - t_0)} U_1(t_0). \tag{A.40}$$

In summary, the solution reads as follows for the special case $f(U) = \lambda U$:

$$U(t, q) = \begin{cases} e^{-\lambda t} U_0(q - vt), & q - vt > 0 \\ e^{-\lambda \frac{q}{v}} U_1\left(t - \frac{q}{v}\right), & q - vt < 0 \end{cases}. \tag{A.41}$$

Remark. Here, initial conditions similar to case (a) could be created artificially, by extending the domain to the upper left quadrant and calculating suitable initial conditions for the negative q-axis. Hence, Eq. (A.37b) has to be replaced by

$$U(0, q) = \tilde{U}_0(q) = e^{\lambda t_0} U_1\left(-\frac{q}{v}\right) \quad \overset{t_0 = 0 - \frac{q}{v}}{=} \quad e^{-\lambda \frac{q}{v}} U_1\left(-\frac{q}{v}\right) \quad \forall q < 0 \tag{A.42}$$

and, thus,

$$\tilde{U}_0(q) = \begin{cases} U_0(q), & q > 0 \\ e^{-\lambda \frac{q}{v}} U_1\left(-\frac{q}{v}\right), & q < 0 \end{cases}. \tag{A.43}$$

Finally, this leads to the solution

$$U(t,q) = e^{-\lambda t}\tilde{U}_0(q - vt). \tag{A.44}$$

A.4.2 Multi-dimensional

Now, the multidimensional case is considered, as required for this thesis.

$$\boldsymbol{U}_t + \boldsymbol{A}\boldsymbol{U}_q = \boldsymbol{M}\boldsymbol{U} \qquad q, t > 0, \tag{A.45a}$$

$$\boldsymbol{U}(0,q) = \boldsymbol{U}_0(q) \qquad \forall q > 0, \tag{A.45b}$$

$$\boldsymbol{U}(t,0) = \boldsymbol{U}_1(t) \qquad \forall t > 0. \tag{A.45c}$$

with $\boldsymbol{U} : \mathbb{R}^2 \to \mathbb{R}^n$ and $\boldsymbol{A}, \boldsymbol{M} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{A} = \operatorname{diag}(v_1, \ldots, v_n)$. Hence, n one-dimensional linear transport equations are coupled via a transition matrix $\boldsymbol{M}$.

Homogenous Solution

The homogenous system

$$\boldsymbol{U}_t + \boldsymbol{A}\boldsymbol{U}_q = \boldsymbol{0} \tag{A.46}$$

constitutes n decoupled transport equations

$$U_{it} + v_i U_{iq} = \boldsymbol{0} \tag{A.47}$$

for $i = 1, \ldots, n$, since $\boldsymbol{A}$ is diagonal. The solutions have been discussed in Section A.4.1. ($f(\boldsymbol{U}) = \boldsymbol{0}$ leads to constant curves along the characteristics.)

$$U_i(t,q) = \begin{cases} U_0(q - v_i t), & q - v_i t > 0 \\ U_1\left(t - \frac{q}{v_i}\right), & q - v_i t < 0 \end{cases} \tag{A.48}$$

Inhomogeneous Solution

Unfortunately, Eq. (A.45) is coupled through the transition matrix $\boldsymbol{M}$. Therefore, the aforementioned strategies are not applicable anymore. In the following it is shown, how the explicit solution proposed in [Xu+16, Eq. (18)] might have been derived and what might have gone wrong in their derivation.

The authors in [Xu+16] have shown that $\boldsymbol{M}$ is diagonalizable in their scenario, i. e.,

$$\boldsymbol{M} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}. \tag{A.49}$$

This can be used to perform a coordinate transformation with the aim to decouple the PDE system (A.45) as follows

$$\frac{\partial}{\partial t}\boldsymbol{U} + \boldsymbol{A}\frac{\partial}{\partial q}\boldsymbol{U} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}\boldsymbol{U} \tag{A.50}$$

$$\boldsymbol{V}^{-1}\frac{\partial}{\partial t}\boldsymbol{U} + \boldsymbol{V}^{-1}\boldsymbol{A}\frac{\partial}{\partial q}\boldsymbol{U} = \boldsymbol{D}\boldsymbol{V}^{-1}\boldsymbol{U} \tag{A.51}$$

If now $\boldsymbol{A}\boldsymbol{V}^{-1} = \boldsymbol{V}^{-1}\boldsymbol{A}$ was true, which is in general not true, then the equation could be reformulated as

$$\frac{\partial \boldsymbol{V}^{-1}\boldsymbol{U}}{\partial t} + \boldsymbol{A}\frac{\partial \boldsymbol{V}^{-1}\boldsymbol{U}}{\partial q} = \boldsymbol{D}\boldsymbol{V}^{-1}\boldsymbol{U}. \tag{A.52}$$

The transformation $\tilde{\boldsymbol{U}} = \boldsymbol{V}^{-1}\boldsymbol{U}$ would then decouple the PDE system with the new coordinates as $\boldsymbol{A}$ and $\boldsymbol{D}$ are diagonal:

$$\frac{\partial \tilde{\boldsymbol{U}}}{\partial t} + \boldsymbol{A}\frac{\partial \tilde{\boldsymbol{U}}}{\partial q} = \boldsymbol{D}\tilde{\boldsymbol{U}} \tag{A.53}$$

or for each component i a simple transport equation

$$\frac{\partial \tilde{\boldsymbol{U}}_i}{\partial t} + v_i\frac{\partial \tilde{\boldsymbol{U}}_i}{\partial q} = \boldsymbol{D}_{ii}\tilde{\boldsymbol{U}}. \tag{A.54}$$

The initial and boundary conditions could be transformed as well into the new coordinate system:

$$\tilde{\boldsymbol{U}}(0,q) = \tilde{\boldsymbol{U}}_0(q) = \boldsymbol{V}^{-1}\boldsymbol{U}(0,q) = \boldsymbol{V}^{-1}\boldsymbol{U}_0(q) \qquad \forall q > 0 \tag{A.55}$$

$$\tilde{\boldsymbol{U}}(t,0) = \tilde{\boldsymbol{U}}_1(t) = \boldsymbol{V}^{-1}\boldsymbol{U}(t,0) = \boldsymbol{V}^{-1}\boldsymbol{U}_1(t) \qquad \forall t > 0 \tag{A.56}$$

or componentwise: $\left(\text{let } \left(\boldsymbol{V}^{-1}\right)_{i,\cdot} \text{ be the } i^{\text{th}} \text{ row of } \boldsymbol{V}^{-1}\right)$

$$\tilde{\boldsymbol{U}}_i(0,q) = \left(\tilde{\boldsymbol{U}}_0\right)_i(q) = \left(\boldsymbol{V}^{-1}\right)_{i,\cdot}\boldsymbol{U}(0,q) = \left(\boldsymbol{V}^{-1}\right)_{i,\cdot}\boldsymbol{U}_0(q) \qquad \forall q > 0 \tag{A.57}$$

$$\tilde{\boldsymbol{U}}_i(t,0) = \left(\tilde{\boldsymbol{U}}_1\right)_i(t) = \left(\boldsymbol{V}^{-1}\right)_{i,\cdot}\boldsymbol{U}(t,0) = \left(\boldsymbol{V}^{-1}\right)_{i,\cdot}\boldsymbol{U}_1(t) \qquad \forall t > 0 \tag{A.58}$$

Accordingly, the solution could be formulated as (cf. (A.41))

$$\tilde{\boldsymbol{U}}_i(t,q) = \begin{cases} e^{-\lambda_i t}\left(\boldsymbol{V}^{-1}\right)_{i,\cdot}\boldsymbol{U}_0(q - v_i t), & q - v_i t > 0 \\ e^{-\lambda_i \frac{q}{v_i}}\left(\boldsymbol{V}^{-1}\right)_{i,\cdot}\boldsymbol{U}_1\left(t - \frac{q}{v_i}\right), & q - v_i t < 0 \end{cases} \tag{A.59}$$

If there were only initial conditions (for $t = 0, q \in \mathbb{R}$) given, the solution would look easier (cf. Eq. (A.36b)):

$$\tilde{\boldsymbol{U}}_i(t,q) = e^{-\lambda_i t} \left(\boldsymbol{V}^{-1}\right)_{i,\cdot} \boldsymbol{U}_0(q - v_i t), \tag{A.60}$$

and the entire solution could be written as

$$\begin{aligned} \boldsymbol{U}(t,q) &= \boldsymbol{V}\tilde{\boldsymbol{U}}(t,q) && \text{(A.61)} \\ &= \boldsymbol{V} \exp\left(\boldsymbol{D}t\right) \boldsymbol{V}^{-1} \boldsymbol{G}(t,q) && \text{(A.62)} \\ &= \exp\left(\boldsymbol{M}t\right) \boldsymbol{G}(t,q) && \text{(A.63)} \end{aligned}$$

with

$$\boldsymbol{G}_i(t,q) = \boldsymbol{U}_0(q - v_i t) \tag{A.64}$$

as it is given in [Xu+16, Eq. (18)].

However, the two flaws leading to Eqs. (A.52) and (A.60) were necessary to obtain this result.

A.4.3 Inserting the incorrect explicit solution into the PDE

To see that the provided explicit solution

$$\boldsymbol{U}(t,q) = \exp(\boldsymbol{M}t) \cdot \boldsymbol{G}(t,q) \tag{A.65}$$

with

$$\boldsymbol{G}(t,q) = \begin{pmatrix} f(g(t,q;v_1)) \\ f(g(t,q;v_2)) \\ \vdots \\ f(g(t,q;v_K)) \end{pmatrix}, \quad f(x) = 1 - \frac{1}{2}\operatorname{erfc}\left(-\frac{x}{\sqrt{2\alpha}}\right), \quad g(t,q;v) = q - vt \tag{A.66}$$

is wrong, it is also possible to insert it into the LHS of the PDE system (A.45), i. e., into

$$\frac{\partial}{\partial t}\boldsymbol{U}(t,q) + \boldsymbol{A}\frac{\partial}{\partial q}\boldsymbol{U}(t,q) \tag{A.67}$$

with $\mathbf{V} = \operatorname{diag}(v_1, \ldots, v_K)$.

Here, the function f is a smoothed step function with a smoothness parameter $\alpha > 0$, as sometimes the explicit solution is provided in this smoothed version. However, this includes the discontinuous case as well by setting $\alpha \to 0$.

With

$$\frac{\partial}{\partial t}\boldsymbol{U}(t,q) = \boldsymbol{M}\exp(\boldsymbol{M}t)\boldsymbol{G}(t,q) + \exp(\boldsymbol{M}q)\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{G}(t,q) \tag{A.68}$$

$$= \boldsymbol{M}\boldsymbol{U}(t,q) + \exp(\boldsymbol{M}t)\boldsymbol{G}_t\boldsymbol{G}'(t,q) \tag{A.69}$$

$$\frac{\partial}{\partial q}\boldsymbol{U}(t,q) = \exp(\boldsymbol{M}t)\frac{\mathrm{d}}{\mathrm{d}q}\boldsymbol{G}(t,q) \tag{A.70}$$

$$= \exp(\boldsymbol{M}t)\boldsymbol{G}_q\boldsymbol{G}'(t,q) \tag{A.71}$$

where

$$\boldsymbol{G}_t := \begin{pmatrix} \frac{\partial}{\partial t}g(t,q;v_1) & & \\ & \ddots & \\ & & \frac{\partial}{\partial t}g(t,q;v_K) \end{pmatrix} = \boldsymbol{A} \tag{A.72}$$

$$\boldsymbol{G}_q := \begin{pmatrix} \frac{\partial}{\partial q}g(t,q;v_1) & & \\ & \ddots & \\ & & \frac{\partial}{\partial q}g(t,q;v_K) \end{pmatrix} = \boldsymbol{I} \tag{A.73}$$

$$\boldsymbol{G}'(t,q) := \begin{pmatrix} f'(g(t,q;v_1)) \\ \vdots \\ f'(g(t,q;v_K)) \end{pmatrix}, \tag{A.74}$$

the LHS yields

$$\frac{\partial}{\partial t}\boldsymbol{U}(t,q) + \boldsymbol{A}\frac{\partial}{\partial q}\boldsymbol{U}(t,q) \tag{A.75}$$

$$= \boldsymbol{M}\boldsymbol{U}(t,q) + \exp(\boldsymbol{M}t)\boldsymbol{G}_t\boldsymbol{G}'(t,q) + \boldsymbol{A}\exp(\boldsymbol{M}t)\boldsymbol{G}_q\boldsymbol{G}'(t,q) \tag{A.76}$$

$$= \boldsymbol{M}\boldsymbol{U}(t,q) + \left[\exp(\boldsymbol{M}t)\boldsymbol{G}_t + \boldsymbol{A}\exp(\boldsymbol{M}t)\boldsymbol{G}_q\right]\boldsymbol{G}'(t,q). \tag{A.77}$$

Thus, to obtain the original equation the last summand has to be zero for all t, q, but inserting $\boldsymbol{G}_t$ and $\boldsymbol{G}_q$ yields

$$\frac{\partial}{\partial t}\boldsymbol{U}(t,q) + \boldsymbol{A}\frac{\partial}{\partial q}\boldsymbol{U}(t,q) = \boldsymbol{M}\boldsymbol{U}(t,q) + \underbrace{\left[-\exp(\boldsymbol{M}t)\boldsymbol{A} + \boldsymbol{A}\exp(\boldsymbol{M}t)\right]}_{\neq \boldsymbol{0},\ \text{since}\ \exp(\boldsymbol{M}t)\boldsymbol{A} \neq \boldsymbol{A}\exp(\boldsymbol{M}t)}\boldsymbol{G}'(t,q). \tag{A.78}$$

If $\boldsymbol{M}$ had only negative eigenvalues, then this could result in a good approximation for high values of t, since the under-braced term then tends to $\boldsymbol{0}$ for $t \to \infty$.

But unfortunately $\boldsymbol{M}$ has always the eigenvalue 0 in our case ($\boldsymbol{M} \cdot \mathbf{1} = \mathbf{0}$), which destroys this hope.

Since $\boldsymbol{G}$ is a smoothed multidimensional step-function, $\boldsymbol{G}'$ is almost zero for many t, q. However, in the transition region close to the steps, i. e., if $q - v_i t \approx 0$ for any i, $\boldsymbol{G}'$ takes high values and even ∞ in the limiting case of a step-function.

A.5 Details on the chosen Finite Volume Method

FVMs (e. g., [LeV07]) are appropriate numerical tools to approximate the solution of transport equations, i. e., first-order PDE systems in space and time that are governed by a conservation law. In this thesis, FVMs are used to solve the system (3.13), which has the following form (cf. Eq. (A.45a) as well).

$$\frac{\partial}{\partial t}\boldsymbol{U}(t,q) + \boldsymbol{A}\frac{\partial}{\partial q}\boldsymbol{U}(t,q) = \boldsymbol{M}\boldsymbol{U}(t,q) \tag{A.79}$$

with initial and boundary conditions

$$\boldsymbol{U}(0,q) = \mathbf{0} \qquad \forall q > 0, \tag{A.80}$$

$$\boldsymbol{U}(t,0) = \mathbf{1} \qquad \lim_{q\to\infty} \boldsymbol{U}(t,q) = \mathbf{0} \qquad \forall t \geq 0, \tag{A.81}$$

respectively.

Therefore, the function $\boldsymbol{U}$ is discretized in time and space at first. Let $t_n = n\Delta t$, $q_k := \left(k + \frac{1}{2}\right)\Delta q$ be such a discretization for chosen grid constants Δt and Δq, which have to fulfill the CFL condition Eq. (3.84). Accordingly, the k^{th} cell (or volume) refers to the interval $\left(q_{k-\frac{1}{2}}, q_{k+\frac{1}{2}}\right) = (k\Delta q, (k+1)\Delta q)$. Furthermore, let $\boldsymbol{U}_k^{(n)} := \boldsymbol{U}(t_n, q_k)$ denote the value of the function at the n^{th} time step of the k^{th} grid cell (cf. Fig. 3.8).

For the inhomogeneous system, a fractional-step method [LeV07, Chapter 17] is implemented as follows. A fractional-step method treats the homogeneous part of the system (A.79), i. e.,

$$\frac{\partial}{\partial t}\boldsymbol{U}(t,q) + \boldsymbol{A}\frac{\partial}{\partial q}\boldsymbol{U}(t,q) = 0, \tag{A.82}$$

and the ODE part

$$\frac{\partial}{\partial t}\boldsymbol{U}(t,q) = \boldsymbol{M}\boldsymbol{U}(t,q) \tag{A.83}$$

separately by applying an FVM method and an ODE solver, respectively.

A general FVM (for a homogeneous PDE like Eq. (A.82)) is defined as (cf. Fig. 3.8 for an illustration)

$$\boldsymbol{U}_k^{(n+1)} := \boldsymbol{U}_k^{(n)} - \frac{\Delta t}{\Delta q}\left(\mathcal{F}_{k+\frac{1}{2}}^{(n)} - \mathcal{F}_{k-\frac{1}{2}}^{(n)}\right), \tag{A.84}$$

where $\mathcal{F}_{k-\frac{1}{2}}^{(n)}$ denotes an approximation of the average flux within the $(n+1)^{\text{th}}$ time step through the cell boundary $q_{k-\frac{1}{2}} = q_k - \frac{1}{2}\Delta q$, i. e.,

$$\mathcal{F}_{k-\frac{1}{2}}^{(n)} \cong \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \boldsymbol{A}\left(\boldsymbol{U}\left(t, q_{k-\frac{1}{2}}\right)\right) \mathrm{d}t. \tag{A.85}$$

As the flux $\mathcal{F}_{k-\frac{1}{2}}^{(n)}$ can be approximated in various ways, there are many different FVM schemes. Eq. (A.84) can be interpreted as the new cell content being the old cell content minus the content that is leaving to (or coming from, if the sign is negative) the neighboring cells within one time step, such that the conservation law is preserved.

A scheme with a slope limiter was chosen for the first part of the fractional-step method used in this thesis. Flux limiters are introduced to avoid oscillations at discontinuities, such that it is possible to benefit from second-order accuracy where the function is smooth without running into numerical problems. Thus, the flux is approximated with a second order formula

$$\mathcal{F}_{k-\frac{1}{2}}^{(n)} := \boldsymbol{A}\boldsymbol{U}_{k-1}^{(n)} + \frac{1}{2}\boldsymbol{A}\left(\Delta q\boldsymbol{I} - \boldsymbol{A}\Delta t\right)\sigma_k^{(n)}, \tag{A.86}$$

where $\sigma_k^{(n)}$ approximates the slope between adjacent grid cells. There are different strategies to determine this slope, e. g., choosing a difference quotient would be the most intuitive and easiest approach. In this thesis, the so called *superbee* slope limiter [LeV07, Section 6.9] for the slope $\sigma_k^{(n)}$ is chosen, to avoid the aforementioned oscillations, which is defined as follows

$$\sigma_k^{(n)} := \operatorname{maxmod}\left(\sigma_k^{(1)}, \sigma_k^{(2)}\right) \tag{A.87}$$

with

$$\sigma_k^{(1)} := \operatorname{minmod}\left(\left(\frac{\boldsymbol{U}_{k+1}^{(n)} - \boldsymbol{U}_k^{(n)}}{\Delta q}\right), 2\left(\frac{\boldsymbol{U}_k^{(n)} - \boldsymbol{U}_{k-1}^{(n)}}{\Delta q}\right)\right), \tag{A.88}$$

$$\sigma_k^{(2)} := \operatorname{minmod}\left(2\left(\frac{\boldsymbol{U}_{k+1}^{(n)} - \boldsymbol{U}_k^{(n)}}{\Delta q}\right), \left(\frac{\boldsymbol{U}_k^{(n)} - \boldsymbol{U}_{k-1}^{(n)}}{\Delta q}\right)\right). \tag{A.89}$$

Herein, the functions minmod and maxmod are defined as

$$\operatorname{minmod}(a,b) := \begin{cases} a & \text{if } |a| \leq |b| \text{ and } ab > 0, \\ b & \text{if } |b| < |a| \text{ and } ab > 0, , \\ 0 & \text{if } ab \leq 0, \end{cases} \tag{A.90}$$

$$\operatorname{maxmod}(a,b) := \begin{cases} a & \text{if } |a| \geq |b| \text{ and } ab > 0, \\ b & \text{if } |b| > |a| \text{ and } ab > 0, , \\ 0 & \text{if } ab \leq 0, \end{cases} \tag{A.91}$$

for $a, b \in \mathbb{R}$, respectively. As the schemes are formulated for the multidimensional case, the operators should be interpreted as componentwise operators.

Putting everything together, the resulting scheme for the intermediate result $\tilde{\boldsymbol{U}}_k^{(n+1)}$ of the homogeneous part can be written as

$$\tilde{\boldsymbol{U}}_k^{(n+1)} = \boldsymbol{U}_k^{(n)} - \boldsymbol{A}\frac{\Delta t}{\Delta q}\left(\boldsymbol{U}_k^{(n)} - \boldsymbol{U}_{k-1}^{(n)}\right) - \frac{1}{2}\boldsymbol{A}\frac{\Delta t}{\Delta q}\left(\Delta q \boldsymbol{I} - \boldsymbol{A}\Delta t\right)\left(\sigma_k^n - \sigma_{k-1}^n\right). \tag{A.92}$$

The ODE part is then handled by an implicit Euler scheme

$$\boldsymbol{U}_k^{(n+1)} = \left(\boldsymbol{I} - \Delta t \boldsymbol{M}\right)^{-1} \tilde{\boldsymbol{U}}_k^{(n+1)}. \tag{A.93}$$

List of Abbreviations

3G	third generation
3GPP	3rd Generation Partnership Project
4G	fourth generation
5G	fifth generation
BBU	base band unit
BS	base station
CA	carrier aggregation
CAPEX	capital expenditure
CBR	constant bitrate
CCDF	complementary cumulative distribution function
CDF	cumulative distribution function
cf.	compare (latin: confer)
CFL	Courant–Friedrichs–Lewy
CI	confidence interval
cMTC	critical machine-type communication
COTS	commercial of the shelf
C-RAN	cloud-RAN or centralized-RAN
D2D	device-to-device
DC	dual connectivity
DES	discrete-event simulation
DHCP	dynamic host configuration protocol
DL	downlink
DoF	degree of freedom
DR	dedicated resources
E2E	end-to-end
e. g.	for example (latin: exempli gratia)
eMBB	enhanced mobile broadband
FFT	fast Fourier transformation
FIFO	first in first out
FTP	file transfer protocol
FVM	finite volume method
GPS	global positioning system
HetNet	heterogeneous network

HPC high performance computing
ICI inter-cell interference
ICMP Internet control message protocol
i. e. that is (latin: id est)
iff if and only if
i.i.d. independent and identically distributed
IoT Internet of things
ITU International Telecom Union
ITS intelligent transport system
KPI key performance indicator
LHS left-hand side
LIFO last in first out
LTE Long-Term Evolution
MAC medium access control
MANO management and orchestration
MBB mobile broadband
MC multi connectivity
MEC mobile edge computing
MGF moment generating function
mMTC massive machine-type communication
MTC machine-type communication
n/a not available
NFV network functions virtualization
NGMN Next Generation Mobile Networks
NR new radio
ODE ordinary differential equation
OPEX operational expenditure
PDCP packet data convergence protocol
PDE partial differential equation
PDF probability density function
PHY physical layer
PLR packet loss rate
PMF probability mass function
PPU packet processing unit
PQwop priority queuing without preemption
PQwp priority queuing with preemption
PS processor sharing
QoE quality of experience

QoS quality of service
RAN radio access network
RAT radio access technology
RB resource block
RLC radio link control
RHS right-hand side
RR round robin
RRC radio resource control
RRU remote radio unit
RTT round trip time
RV random variable
SCS sub carrier spacing
SDN software-defined network
SINR signal-to-interference-plus-noise ratio
SLA service level agreement
SON self-organizing network
TTI transmission time interval
TSN time-sensitive network
UE user equipment
UL uplink
uMTC ultra-reliable machine-type communication
URLLC ultra-reliable low latency communication
V2V vehicle-to-vehicle
VBR variable bitrate
VoIP voice over Internet protocol
WFQ weighted fair queuing
w.p. with probability
WRED weighted random early discard

List of Symbols

Units of measure

bit	Bit
bps	Bits per second
B	Byte
dB	Decibel
dBi	Decibel (isotropic)
dBm	Decibel (related to 10^{-3} W)
h	Hour
Hz	Hertz ($1\mathrm{Hz} = 1\mathrm{s}^{-1}$)
m	Meter
s	Second
W	Watt

Constants

k_B	Boltzmann constant, $k_\mathrm{B} = 1.3806488 \cdot 10^{-23}\,\frac{\mathrm{J}}{\mathrm{K}}$
e	Euler's number, $e = 2.71828\ldots$

Number Fields

$\mathbb{N}$	Natural numbers, $\mathbb{N} := \{0, 1, 2, \ldots\}$
$\mathbb{R}$	Real numbers
$X_{>0}$	X restricted to positive numbers, $X_{>0} := X \cap (0, \infty)$
$X_{\geq 0}$	X restricted to non-negative numbers, $X_{\geq 0} := X \cap [0, \infty)$

Operators and Functions

$!$	Factorial, $n! = \prod_{i=1}^{n} i$
$*$	Convolution operator

$\times$	Cartesian product
$\stackrel{d}{=}$	Equality in distribution
$\|\cdot\|$	Norm
$(\cdot)^T$	Transpose operator
$\frac{\mathrm{d}}{\mathrm{d}x}$	Derivation operator with respect to x
$\frac{\partial}{\partial x}$	Partial derivation operator with respect to x
$\in$	Element relation
δ_{ij}	Kronecker Delta
$\delta(\cdot)$	Dirac impuls at zero
φ	Fixed point operator
ϕ	Operator that replaces the negative part of a PDF by a dirac impulse at zero
Φ	Operator that cuts the negative part of a CDF by setting it to zero
$\chi_A(\cdot)$	Indicator function of a set A
argmin	Operator which selects the parameter that minimizes an expression
$\mathrm{diag}(\mathbf{x})$	Diagonal matrix with the vector $\mathbf{x}$ on the diagonal
erfc	Error function
maxmod	Operator that selects the argument with the larger modulus (cf. Eq. (A.90))
minmod	Operator that selects the argument with the smaller modulus (cf. Eq. (A.91))
sgn	Sign function
$d(\cdot,\cdot)$	Metric
$\mathbb{E}[\cdot]$	Expectation operator
f_X	PDF of the RV X
F_X	CDF of the RV X
M_X	MGF of the RV
$\mathbb{P}[\cdot]$	Probability operator

Variables and Symbols

$\emptyset$	Empty set, $\emptyset := \{\}$
$\mathbf{0}$	Column vector or matrix of zeros of suitable dimension
$\mathbf{1}$	Column vector of ones of suitable dimension
∞	Infinity
α	Overall arrival rate to a queuing network
γ	SINR

$\zeta_i(\boldsymbol{y})$	Probability that an active flow in cell i experiences the interference scenario $\boldsymbol{y}$
κ	Length of a path in a queuing network
$\boldsymbol{\Lambda}$	Matrix containing transition rates between interference scenarios
$\boldsymbol{\lambda}$	Vector of arrival rates
λ	Mean arrival rate to a queue
λ_{LTE}	Mean arrival rate to an LTE node
$\lambda_{\boldsymbol{M}}$	Eigenvalue of a matrix
λ_{NR}	Mean arrival rate to an NR node
μ	Mean service rate of a queue
μ_{LTE}	Mean service rate of an LTE node
μ_{NR}	Mean service rate of an NR node
μ_{PP}	Mean service rate of the packet processing
ξ	RV of the file size
$(\boldsymbol{\xi}, \eta_i)$	Target state for the definition of $\boldsymbol{V}_{\boldsymbol{u},i}$
$\boldsymbol{\Pi}$	Joint probability of observed competing flows and interference
π	State probability
ρ	Queue load
ρ_{LTE}	Load of an LTE node
ρ_{NR}	Load of an NR node
$\boldsymbol{\Upsilon}$	RV of user locations in all cells
$\boldsymbol{\upsilon}$	User locations in all cells
Ψ	Sum of all rates leaving a state
$\tilde{\Psi}$	Sum of all rates leaving a state with absorbing state $\mathbf{A}$
ψ	Rate to the absorbing state
Ω	Mean file size
$\mathbf{A}$	Absorbing state
$\boldsymbol{A}$	Diagonal matrix containing all download (transport) rates
A	Distribution of the inter-arrival time
$\mathcal{A}_t$	RV of the number of arrivals within time t
$\boldsymbol{B}$	Diagonal matrix containing all effective download rates
$\bar{\mathcal{B}}$	Closed circle
B	Distribution of the service time

B_{BW}	Bandwidth
C	Distribution of the breathing time
$\mathscr{C}$	Computational complexity
$\mathcal{C}$	Set of traffic classes
$\mathfrak{c}$	Traffic class index
c	Number of servers of a queue
c	Achievable rate
$c_{\max}$	Maximum achievable data rate
$\mathcal{D}$	Queuing discipline
D	Additional (technical) delay
$\boldsymbol{D_M}$	Diagonal transformation matrix
d_j	Distance of a UE to BS j
e_{BW}	Bandwidth efficiency
$\boldsymbol{e}_i$	Column unit vector of suitable dimension with $(\boldsymbol{e}_i)_j = \delta_{ij}$
e_{SINR}	SINR efficiency
$\mathcal{F}$	Flux between finite volumes
f_c	Carrier frequency
$f_{\boldsymbol{v},-i}$	Joint density of users in neighbor cells
$f_{\boldsymbol{v},i,\boldsymbol{y}}$	Joint density of users in active neighbor cells
$\mathcal{G}$	Gershgorin circle
G	Gain
G_{BS}	BS noise figure
G_{div}	Antenna diversity gain
G_{FF}	Fading margin
G_{UE}	UE noise figure
g	Function of the target KPI to optimize
h	Height
h_{BS}	BS height
h_{UE}	UE height
$\boldsymbol{I}$	Identity matrix of suitable dimension, $\boldsymbol{I}_{ij} = \delta_{ij}$
$\boldsymbol{\mathcal{I}}$	RV of the observed states by the tagged flow
i	Index referring to a cell or queue
J	Sojourn time of a queue
$\tilde{J}$	Latency (Sojourn time of a queue plus technical delays)

$\boldsymbol{J}_M, \boldsymbol{J}_l$	Jordan matrix, Jordan Block
j	Index referring to a cell or queue
K	Capacity of a queue
k	Index variable
k_{SCS}	Factor of the SCS numerology
L	Size of the neighbor's interference space. $L = \lvert\mathcal{Y}_i\rvert$
$\mathcal{L}$	Considered Region in $\mathbb{R}^2$
$\mathcal{L}_i$	Area of cell i
$\mathscr{L}$	Cartesian product of all cell regions
$\mathscr{L}_{i,\boldsymbol{y}}$	Cartesian product of active cell regions
$\mathscr{L}_i(\boldsymbol{u})$	Cartesian product of all cell regions with the i^{th} component fixed to $\boldsymbol{u}$
l	Index variable
l_{path}	Path loss
$\boldsymbol{M}$	General transition matrix
$\boldsymbol{M}_{\boldsymbol{u},i}$	Matrix containing all possible state transitions
$\tilde{\boldsymbol{M}}_{\boldsymbol{u},i}$	Modified transition matrix after introducing absorbing state
$\boldsymbol{M}^{\boldsymbol{V}}_{\boldsymbol{u},i}$	Transition matrix incorporating download rates
$\boldsymbol{M}^{\boldsymbol{W}}_{\boldsymbol{u},i}$	Transition matrix incorporating effective download rates
$\boldsymbol{M}_{\mathrm{in}}$	Matrix containing the inner transitions
M	Number of queues in a queuing network
$\mathcal{M}$	Index set for the queues in a queuing network
m	Index variable
$\boldsymbol{N}$	The non-diagonal (nilpotent) part of a Jordan block
N	Number of cells
N_0	Spectral density of the thermal noise power
N_{F}	Number of simulated flows
N_{G}	Number of discretized cells
N_{iter}	Number of iterations
N_Q	Population of a queue
N_{t}	Number of time steps
$N_{\boldsymbol{u}}$	Number of locations
$\mathcal{N}$	Index set for the cells in a mobile network
$\mathcal{N}_{-i}$	Index set of cell i's neighbors
$\mathcal{N}_{i,1}, \mathcal{N}_{i,0}$	Index sets of cell i's active and inactive neighbors
n	Index variable in time
$\bar{n}$	Mean number of active flows in a queue

$\mathcal{O}$	Landau O notation, $\mathcal{O}(f(x))$ can be upper-bounded by $cf(x)$ for a constant $c \in \mathbb{R}_{>0}$ and $x \to \infty$
$\boldsymbol{P}$	Routing matrix
$\tilde{\boldsymbol{P}}$	Extended routing matrix
$\boldsymbol{P}_M$	Jordan transformation matrix
$\mathcal{P}$	Set of parameters for optimization
P^{b}	Blocking probability
P^{st}	Starvation probability
$\boldsymbol{p}_{0\cdot}$	Arrival probabilities
$\boldsymbol{p}_{\cdot 0}$	Departure probabilities
p^{rx}	Received power
$\bar{p}^{\mathrm{rx}}$	Power control value
Q	Video Buffer
$\boldsymbol{q}$	A path in a queuing network
q_{a}	Buffer threshold
R	Random remaining fraction
$R_{\mathfrak{c}}$	Virtual rate of class $\mathfrak{c}$
R_{CBR}	The constant video bitrate
R_{cell}	Cell throughput
R_{flow}	Flow throughput
r	The number of a percentile
r_{req}	The number of a percentile in a constraint
S	RV of the service time
$\bar{S}$	Mean of the service time
$\mathcal{S}_k$	RV of the sum of k independent service times
$s_n^{(\mathfrak{c})}$	Service time of the n^{th} object of class $\mathfrak{c}$
T	RV of the inter-arrival time
$\bar{T}$	Mean of the inter-arrival time
$\mathcal{T}_k$	RV of the sum of k independent inter-arrival times
T_{cp}	Computation time
T_{center}	Traffic demand in the considered cell
T_{neighbor}	Traffic demand in the neighboring cells
T_{out}	Resulting RV of the inter-arrival times after a queue operation
T_{real}	Simulated real time
T_{TTI}	Length of one TTI

T_{video}	RV of the video duration
$\bar{T}_{\text{video}}$	Mean of the video duration
T_W	Argument for the additional waiting time $\hat{W}$
t_{a}	Time to prefetch the buffer (startup delay)
$t_{\text{arr},n}^{(\mathfrak{c})}$	Arrival time of the n^{th} object of class $\mathfrak{c}$
t_{async}	Upper bound for asynchronous mMTC arrivals
$t_{\text{dep},n}^{(\mathfrak{c})}$	Departure time of the n^{th} object of class $\mathfrak{c}$
t_{req}	Latency constraint
t_{sync}	Upper bound for asynchronous mMTC arrivals
$t_{\text{vs},n}^{(\mathfrak{c})}$	Virtual start time of the n^{th} object of class $\mathfrak{c}$
$t_{\text{vf},n}^{(\mathfrak{c})}$	Virtual finish time of the n^{th} object of class $\mathfrak{c}$
tol	Tolerance for the algorithm to stop
U	RV of the breathing time
$\boldsymbol{U}_{\boldsymbol{u},i}$	Startup delay distribution vector for all states
$\bar{U}_{\boldsymbol{u},i}$	Startup delay distribution at one location $\boldsymbol{u}$
$\bar{U}_i$	Startup delay distribution at one location over the entire cell
$U_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$	Component of $\boldsymbol{U}_{\boldsymbol{u},i}$ for one state $(\boldsymbol{y}, z_i)$
$\boldsymbol{u}$	User location
$\boldsymbol{u}_{\text{edge}}$	A cell edge location
$\boldsymbol{V_M}$	Diagonalization transformation matrix
$\boldsymbol{V}_{\boldsymbol{u},i}$	Matrix containing probabilities that prefetching ends in a certain state $(\boldsymbol{\xi}, \eta_i)$, when it started in a certain state $(\boldsymbol{y}, z_i)$
$V_{(\boldsymbol{y},z_i)}^{(\boldsymbol{\xi},\eta_i)}$	Components of $\boldsymbol{V}_{\boldsymbol{u},i}$
$v_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$	Download rate at location $\boldsymbol{u}$ in cell i and state $(\boldsymbol{y}, z_i)$
W	Waiting time of a queue
$\hat{W}^{(\mathfrak{c})}$	Additional waiting time due to class $\mathfrak{c}$ arrivals while waiting
$\boldsymbol{W}_{\boldsymbol{u},i}$	Starvation probability vector for a given buffer
$W^{(\boldsymbol{y},z_i)}$	Component of $\boldsymbol{W}_{\boldsymbol{u},i}$
$w_{\mathfrak{c}}$	Scheduler weight of class $\mathfrak{c}$
$w_{\boldsymbol{u},i}^{(\boldsymbol{y},z_i)}$	Effective Download rate at location $\boldsymbol{u}$ in cell i and state $(\boldsymbol{y}, z_i)$
$\boldsymbol{X}$	Random process of the joint queue state
$\mathcal{X}$	Joint state space of all queues
$\boldsymbol{x}$	Realization of the joint queue state
$\boldsymbol{Y}$	Random process of interference
$\tilde{\boldsymbol{Y}}$	Random process of interference with BS i

$\mathcal{Y}$	Joint state space of BS activities
$\mathcal{Y}_i$	Neighbor interference space of BS i, i. e., $\mathcal{Y}$ restricted to scenarios where BS i is active
$\boldsymbol{y}$	Interference scenario (realization of $\boldsymbol{Y}$)
$(\boldsymbol{y}, z_i)$	Joint state of interference scenario and observed other flows
$\mathcal{Z}_i^{\mathrm{PB}}$	State Space of competing flows at BS i during playback
Z_i^{PB}	RV of competing flows at BS i during playback
$\mathcal{Z}_i^{\mathrm{PF}}$	State Space of competing flows at BS i during prefetching
Z_i^{PF}	RV of competing flows at BS i during prefetching
z_i	Realization of the competing flows state

Distributions and Stochastic Processes

$\mathrm{B}(p, q)$	Beta distribution with shape parameters p, q
$\Gamma(\alpha, \beta)$	Gamma distribution with shape parameter α and rate parameter β
D	Deterministic process
$\mathrm{Exp}(\lambda)$	Exponential distribution with rate parameter λ
GI	General independent random process
$\mathrm{Geo}(p)$	Geometric distribution with success probability p
M	Markov process
$\mathcal{U}(a, b)$	Uniform distribution on the interval $[a, b]$

List of Figures

List of Tables

List of Algorithms

Bibliography

[3GP13a] 3GPP. *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Study on Small Cell Enhancements for E-UTRA and E-UTRAN – Higher layer aspects (Release 12)*. Tech. Report TR 36.842 V1.0.0. Nov. 2013 (cit. on p. 72).

[3GP18a] 3GPP. *NR; NR and NG-RAN overall description; stage 2 (Release 15)*. Tech. Specification TS 38.300 NR. 2018 (cit. on p. 75).

[3GP11a] 3GPP. *Study on RAN Improvements for Machine-type communications*. Tech. Report TR 37.868 V1.1.0. 3GPP, Aug. 2011 (cit. on pp. 14–16).

[3GP10] 3GPP. *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) - Further Advancements for E-UTRA: Physical Layer Aspects*. Tech. Report TR 36.814 V9.0.0. 2010 (cit. on pp. 34, 55, 113).

[3GP11b] 3GPP. *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 10)*. Tech. Specification TR 36.300 V10.6.0. Dec. 2011 (cit. on p. 1).

[3GP08a] 3GPP. *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 8)*. Tech. Specification TR 36.300 V8.7.0. Dec. 2008 (cit. on p. 1).

[3GP13b] 3GPP. *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Carrier Aggregation; Base Station (BS) radio transmission and reception (Release 10)*. Tech. Report TR 36.808 V10.1.0. July 2013 (cit. on p. 72).

[3GP18b] 3GPP. *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios (Release 15)*. Tech. Report TR 36.942 V15.0.0. 3GPP, June 2018 (cit. on p. 113).

[3GP08b] 3GPP. *Technical Specification Group Radio Access Network; Feasibility study for evolved Universal Terrestrial Radio Access (UTRA) and Universal Terrestrial Radio Access Network (UTRAN) (Release 8)*. Tech. Report TR 25.912 V8.0.0. Dec. 2008 (cit. on p. 8).

[3GP17] 3GPP. *Technical Specification Group Radio Access Network; Study on New Radio Access Technology; Physical Layer Aspects; (Release 14)*. Tech. Report TR 38.802 V14.2.0. 3GPP, Sept. 2017 (cit. on p. 16).

[3GP11c] 3GPP. *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions (Release 9)*. Tech. Report TR 36.902 V9.3.1. 3GPP, Mar. 2011 (cit. on p. 99).

[5GP15] 5GPPP. *Automotive Vision*. White Paper. Aug. 2015 (cit. on p. 4).

[Abe10] S. Abeta. "Toward LTE commercial launch and future plan for LTE enhancements (LTE-Advanced)". In: *2010 IEEE International Conference on Communication Systems*. Nov. 2010, pp. 146–150 (cit. on p. 8).

[ALB16] H. Al-Zubaidy, J. Liebeherr, and A. Burchard. "Network-Layer Performance Analysis of Multihop Fading Channels". In: *IEEE/ACM Transactions on Networking* 24.1 (Feb. 2016), pp. 204–217 (cit. on p. 8).

[Asm03] S. Asmussen. *Applied probability and queues*. 2. ed. New York , Berlin , Heidelberg [u.a.]: Springer, 2003 (cit. on pp. 8, 78).

[Bas+75] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios. "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers". In: *J. ACM* 22 (Apr. 1975), pp. 248–260 (cit. on p. 22).

[Ber+14] S. Berger, A. Fehske, P. Zanier, I. Viering, and G. Fettweis. "Comparing Online and Offline SON Solutions for Concurrent Capacity and Coverage Optimization". In: *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. Sept. 2014, pp. 1–6 (cit. on p. 99).

[Ber15] S. Berger. "Simultane Downlink und Uplink Selbstorganisation der Antennenneigungswinkel zur Verbesseerung von Datendurchsatz und Netzabdeckung". Dissertation. PhD thesis. TU Dresden, 2015 (cit. on p. 99).

[BG92] D. Bertsekas and R. G. Gallager. *Data Networks*. Second. Englewood Cliffs, NJ: Prentice Hall, 1992 (cit. on pp. 23, 25).

[BBP04] T. Bonald, S. C. Borst, and A. Proutiere. "How mobility impacts the flow-level performance of wireless data systems". In: *IEEE INFOCOM 2004*. Vol. 3. Mar. 2004, 1872–1881 vol.3 (cit. on pp. 9, 22, 60, 71).

[Bon04b] T. Bonald, S. Borst, N. Hegde, and A. Proutiére. “Wireless Data Performance in Multi-cell Scenarios”. In: *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*. SIGMETRICS ’04/Performance ’04. New York, NY, USA: ACM, June 2004, pp. 378–380 (cit. on pp. 9, 22, 35, 60, 71).

[Bon05] T. Bonald. “Flow-level performance analysis of some opportunistic scheduling algorithms”. In: *European Transactions on Telecommunications* (Jan. 2005) (cit. on pp. 9, 22, 34, 60).

[Bri+16] B. Briscoe, A. Brunstrom, A. Petlund, et al. “Reducing Internet Latency: A Survey of Techniques and Their Merits”. In: *IEEE Communications Surveys Tutorials* 18.3 (Nov. 2016), pp. 2149–2196 (cit. on p. 5).

[Cab12] S. Caban. *Evaluation of HSDPA and LTE from testbed measurements to system level performance*. Hoboken, N.J.: Wiley, 2012 (cit. on p. 11).

[Cal81] S. Calo. “Message Delays in Repeated-Service Tandem Connections”. In: *IEEE Trans. on Communications* 29.5 (May 1981), pp. 670–678 (cit. on p. 24).

[Cha+19] J. K. Chaudhary, A. Kumar, J. Bartelt, and G. Fettweis. “C-RAN Employing xRAN Functional Split: Complexity Analysis for 5G NR Remote Radio Unit”. In: *2019 European Conference on Networks and Communications (EuCNC)*. June 2019, pp. 580–585 (cit. on p. 73).

[Cis17] Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016 - 2021*. White Paper. White Paper. Cisco, 2017 (cit. on p. 38).

[Col01] S. Coles. *An Introduction to Statistical Modeling of Extreme Values*. London, 2001 (cit. on p. 15).

[CSP16] G. Corrales Madueño, Č. Stefanović, and P. Popovski. “Reliable and Efficient Access for Alarm-Initiated and Regular M2M Traffic in IEEE 802.11ah Systems”. In: *IEEE Internet of Things Journal* 3.5 (Oct. 2016), pp. 673–682 (cit. on p. 14).

[Cro32] C. D. Crommelin. “Delay probability formulas when the holding times are constant”. In: *The Post office electrical engineers’ journal* 25 (1932), pp. 41–50 (cit. on p. 81).

[DN97] C. Dietrich and G. Newsam. “Fast and Exact Simulation of Stationary Gaussian Processes through Circulant Embedding of the Covariance Matrix”. In: *SIAM Journal on Scientific Computing* 18.4 (1997), pp. 1088–1107 (cit. on p. 18).

[EE10] A. El Falou and S. E. Elayoubi. “Uplink Flow Level Capacity for HSPA+ Systems”. In: *2010 IEEE 71st Vehicular Technology Conference*. May 2010, pp. 1–5 (cit. on p. 60).

[Ell+16] F. Ellinger, T. Meister, P. Grosa, et al. “Project FAST - fast actuators sensors transceivers”. In: *2016 IEEE MTT-S Latin America Microwave Conference (LAMC)*. Dec. 2016, pp. 1–3 (cit. on p. 72).

[ETS09] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions*. Tech. Report TR 102 638 V1.1.1. June 2009 (cit. on p. 4).

[Far02] G. E. Farin. *Handbook of computer aided geometric design*. 1. ed. Amsterdam [et.al.] , New York, NY: Elsevier, 2002 (cit. on p. 89).

[FRF09] A. J. Fehske, F. Richter, and G. P. Fettweis. “Energy Efficiency Improvements through Micro Sites in Cellular Mobile Radio Networks”. In: *2009 IEEE Globecom Workshops*. Nov. 2009, pp. 1–5 (cit. on p. 113).

[FF12] A. Fehske and G. Fettweis. “Aggregation of variables in load models for interference-coupled cellular data networks”. In: *Communications (ICC), 2012 IEEE International Conference on*. June 2012, pp. 5102–5107 (cit. on p. 35).

[Fid06] M. Fidler. “WLC15-2: A Network Calculus Approach to Probabilistic Quality of Service Analysis of Fading Channels”. In: *IEEE Globecom 2006*. Nov. 2006, pp. 1–6 (cit. on p. 8).

[FJ93] S. Floyd and V. Jacobson. “Random early detection gateways for congestion avoidance”. In: *IEEE/ACM Trans. Netw.* 1.4 (Aug. 1993), pp. 397–413 (cit. on p. 74).

[Fra01] G. Franx. “A simple solution for the M/D/c waiting time distribution”. In: *Operations Research Letters* 29.5 (2001), pp. 221–229 (cit. on p. 83).

[Fro+14] A. Frotzscher, U. Wetzker, M. Bauer, et al. “Requirements and current solutions of wireless communication in industrial automation”. In: *2014 IEEE International Conference on Communications Workshops (ICC)*. June 2014, pp. 67–72 (cit. on p. 4).

[Ger31] S. A. Gershgorin. “Über die Abgrenzung der Eigenwerte einer Matrix”. In: *Bulletin de l’Académie des Scienses de l’URSS. Classe des sciences mathématiques et naturelles* 8 (1931), pp. 749–754 (cit. on p. 49).

[Gre26] M. Greenwood. “A Report on the Natural Duration of Cancer.” In: *Reports on Public Health and Medical Subjects* 33 (1926) (cit. on p. 83).

[Ham14] F. Hameed Mir Zeeshan and Filali. “LTE and IEEE 802.11p for vehicular networking: a performance evaluation”. In: *EURASIP Journal on Wireless Communications and Networking* 2014.1 (May 2014), p. 89 (cit. on p. 4).

[IEC13] IEC. *Communication networks and systems for power utility automation - Part 3: General requirements*. Norm IEC 61850-3:2013. Dec. 2013 (cit. on p. 4).

[ITU15] ITU-R. *IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond*. Rec. ITU-R M.2083-0. Sept. 2015 (cit. on pp. 1, 2).

[ITU17] ITU-R. *Minimum requirements related to technical performance for IMT-2020 radio interface(s)*. Rep. ITU-R M.2410-0. 2017 (cit. on p. 3).

[ITU08] ITU-R. *Requirements related to technical performance for IMT-Advanced radio interface(s)*. Rep. ITU-R M.2134. 2008 (cit. on p. 1).

[Jac57] J. R. Jackson. "Networks of Waiting Lines". In: *Operations Research* 5.4 (Feb. 1957), pp. 518–521 (cit. on pp. 22, 27).

[Jan11] B. Jann. *Einführung in die Statistik*. 2., bearb. Aufl. München [u.a.]: Oldenbourg, 2011 (cit. on p. 17).

[Jar+11] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia. "Modeling and performance evaluation of an OpenFlow architecture". In: *2011 23rd International Teletraffic Congress (ITC)*. Sept. 2011, pp. 1–7 (cit. on pp. 9, 27, 71).

[Kal60] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.1 (1960) (cit. on p. 100).

[KM58] E. L. Kaplan and P. Meier. "Nonparametric Estimation from Incomplete Observations". In: *Journal of the American Statistical Association* 53.282 (1958), pp. 457–481 (cit. on p. 83).

[Kel75] F. P. Kelly. "Networks of Queues with Customers of Different Types". In: *Journal of Applied Probability* 12.3 (1975), pp. 542–554 (cit. on p. 22).

[Ken53] D. G. Kendall. "Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain". In: *Ann. Math. Statist.* 24.3 (Sept. 1953), pp. 338–354 (cit. on p. 21).

[Kho+18] B. Khodapanah, A. Awada, I. Viering, D. Oehmann, M. Simsek, and G. P. Fettweis. "Fulfillment of Service Level Agreements via Slice-Aware Radio Resource Management in 5G Networks". In: *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. June 2018, pp. 1–6 (cit. on p. 74).

[Kin64] J. F. C. Kingman. "A martingale inequality in the theory of queues". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 60.2 (1964), pp. 359–361 (cit. on pp. 81, 116).

[Kle75] L. Kleinrock. *Queueing systems. 1, Theory*. New York: Wiley, 1975 (cit. on pp. 8, 19, 36, 78).

[Kle76] L. Kleinrock. *Queueing systems. 2, Computer applications*. New York: Wiley, 1976 (cit. on p. 8).

[KFF14] H. Klessig, A. Fehske, and G. Fettweis. “Admission control in interference-coupled wireless data networks: A queuing theory-based network model”. In: *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. May 2014, pp. 151–158 (cit. on pp. 9, 22, 36, 60).

[KF15] H. Klessig and G. Fettweis. “Impact of Inter-Cell Interference on Buffered Video Streaming Startup Delays”. In: *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. Sept. 2015, pp. 1–2 (cit. on pp. 9, 22, 39, 54, 60).

[KGF14] H. Klessig, M. Günzel, and G. Fettweis. “Increasing the Capacity of Large-Scale HetNets through Centralized Dynamic Data Offloading”. In: *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. Sept. 2014, pp. 1–7 (cit. on pp. 9, 22, 60).

[Kle+16] H. Klessig, D. Öhmann, A. J. Fehske, and G. P. Fettweis. “A Performance Evaluation Framework for Interference-Coupled Cellular Data Networks”. In: *IEEE Trans. Wireless Commun.* 15.2 (Feb. 2016), pp. 938–950 (cit. on pp. 9, 17, 22, 31, 33, 35, 36, 55, 60, 62, 113).

[KS12] S. S. Krishnan and R. K. Sitaraman. “Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-experimental Designs”. In: *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*. IMC ’12. Boston, Massachusetts, USA: ACM, 2012, pp. 211–224 (cit. on pp. 38, 59).

[Küh76] P. J. Kühn. “Analysis of Complex Queuing Networks by Decomposition”. In: *8th INternational Teletraffic Congress*. Melbourne, Australia, Nov. 1976 (cit. on pp. 9, 27, 91).

[Küh15] P. J. Kühn. “Real Time Control in 5G: Embedded Communication Networks – A System Theoretic Modeling Approach”. In: *VDE/ITG Section 5.2.4 Workshop - 5G System Architecture*. Munich, Germany, Dec. 2015 (cit. on pp. 9, 27).

[Kun16] H. Kuntzschmann. “Analyse der räumlich-zeitlich verteilten Datenverkehrsnachfrage und ihres Einflusses auf die Latenz im multi-zellularen Kontext”. Diploma Thesis. MA thesis. TU Dresden, 2016 (cit. on p. 16).

[Lan+13a] M. Laner, J. Fabini, P. Svoboda, and M. Rupp. “End-to-end Delay in Mobile Networks: Does the Traffic Pattern Matter?” In: *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*. Aug. 2013, pp. 1–5 (cit. on p. 13).

[Lan+13b] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp. “Traffic Models for Machine Type Communications”. In: *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*. Aug. 2013, pp. 1–5 (cit. on pp. 13, 15).

[Lan+12] M. Laner, P. Svoboda, S. Schwarz, and M. Rupp. “Users in cells: A data traffic analysis”. In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. Apr. 2012, pp. 3063–3068 (cit. on p. 13).

[LT04] J.-Y. Le Boudec and P. Thiran. “Network Calculus: A Theory of Deterministic Queuing Systems for the Internet”. In: (June 2004) (cit. on p. 8).

[Lee+14] D. Lee, S. Zhou, X. Zhong, Z. Niu, X. Zhou, and H. Zhang. “Spatial modeling of the traffic density in cellular networks”. In: *IEEE Wireless Communications* 21.1 (Feb. 2014), pp. 80–88 (cit. on p. 18).

[Lei+11] Lei Zhang, Xin Chen, Xudong Xiang, and Jianxiong Wan. “A stochastic network calculus approach for the end-to-end delay analysis of LTE networks”. In: *2011 International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*. Oct. 2011, pp. 30–35 (cit. on p. 8).

[LeV07] R. J. LeVeque. *Finite Volume Methods For Hyperbolic Problems*. Cambridge: Cambridge Univ. Press, 2007 (cit. on pp. 52, 68, 124, 125).

[Lin52] D. V. Lindley. “The theory of queues with a single server”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 48.2 (1952), pp. 277–289 (cit. on p. 79).

[LJ08] Y. Liu and Y. Jiang. *Stochastic Network Calculus*. Springer, London, 2008 (cit. on p. 8).

[Mah+15] K. Mahmood, A. Chilwan, O. Østerbø, and M. Jarschel. “Modelling of OpenFlow-based software-defined networks: the multiple node case”. In: *IET Networks* 4.5 (Sept. 2015), pp. 278–284 (cit. on pp. 9, 27).

[Mah+14] K. Mahmood, A. Chilwan, O. N. Østerbø, and M. Jarschel. “On The Modeling of OpenFlow-based SDNs: The Single Node Case”. In: *Computer science & information technology : (CS & IT)*. Vol. 4. 11. Nov. 2014 (cit. on pp. 9, 27, 71).

[Mat19] Mathworks. *MATLAB*. 2019. URL: `https://www.mathworks.com/products/matlab.html` (visited on June 1, 2019) (cit. on pp. 9, 12, 28, 109).

[MET16] METIS II. *Refined scenarios and requirements, consolidated use cases, and qualitative techno-economic feasibility assessment*. Rep. METIS-II/D1.1 v.1.0. Jan. 2016 (cit. on p. 2).

[Mog+07] P. Mogensen, W. Na, I. Kovacs, et al. “LTE Capacity Compared to the Shannon Bound”. In: *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*. Apr. 2007, pp. 1234–1238 (cit. on p. 34).

[ML78] C. Moler and C. V. Loan. “Nineteen Dubious Ways to Compute the Exponential of a Matrix”. In: *SIAM Review* 20.4 (1978), pp. 801–836 (cit. on p. 68).

[Nag18] M. Nagireddy. “Cisco Catalyst 9500 High Performance Switch Architecture”. TechWiseTV Workshop. July 2018 (cit. on p. 74).

[Nak05] K. Nakagawa. “On the series expansion for the stationary probabilities of an M/D/1 queue”. In: *Journal of the Operations Research Society of Japan* 2 (June 2005) (cit. on p. 83).

[NGM16] NGMN. *Description of Network Slicing Concept*. Final Deliverable 1.0. Jan. 2016 (cit. on p. 7).

[NGM15] NGMN. *NGMN 5G White Paper*. White Paper 1.0. Jan. 2015 (cit. on p. 6).

[NGM08a] NGMN. *NGMN Radio Access Performance Evaluation Methodology*. White Paper 1.0. Jan. 2008 (cit. on pp. 13, 15, 16, 27).

[NGM08b] NGMN. *NGMN Use Cases related to Self Organising Network, Overall Description*. White Paper 2.02. Dec. 2008 (cit. on p. 99).

[Nik+13] N. Nikaein, M. Laner, K. Zhou, et al. “Simple Traffic Modeling Framework for Machine Type Communication”. In: *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*. Aug. 2013, pp. 1–5 (cit. on p. 13).

[PP02] A. Papoulis and S. U. Pillai. *Probability, random variables, and stochastic processes*. 4. ed., internat. ed. Boston, Mass. [u.a.]: McGraw-Hill, 2002 (cit. on pp. 13, 14).

[PG93] A. K. Parekh and R. G. Gallager. “A generalized processor sharing approach to flow control in integrated services networks: the single-node case”. In: *IEEE/ACM Transactions on Networking* 1.3 (June 1993), pp. 344–357 (cit. on p. 88).

[Pra74] N. U. Prabhu. “Wiener-Hopf Techniques in Queueing Theory”. In: *Mathematical Methods in Queueing Theory*. Ed. by A. B. Clarke. Berlin, Heidelberg: Springer Berlin Heidelberg, 1974, pp. 81–90 (cit. on p. 79).

[Rob01] J. Roberts. “Traffic Theory and the Internet”. In: *IEEE Commun. Mag.* 39.1 (Jan. 2001), pp. 94–99 (cit. on p. 12).

[SGA15] S. Schiessl, J. Gross, and H. Al-Zubaidy. “Delay Analysis for Wireless Fading Channels with Finite Blocklength Channel Coding”. In: *Proc. of the 18th ACM MSWiM ’15*. Cancun, Mexico: ACM, 2015, pp. 13–22 (cit. on pp. 5, 8).

[Sch16] L. Schwartz. “Analyse und Modellierung der räumlich-zeitlich verteilten Datenverkehrsnachfrage in Mobilfunknetzen auf Basis Sozialer Netzwerke”. Diploma Thesis. TU Dresden, 2016 (cit. on pp. 16–18).

[Sha49] C. E. Shannon. “Communication in the Presence of Noise”. In: *Proceedings of the IRE* 37.1 (Jan. 1949), pp. 10–21 (cit. on p. 34).

[SYQ17] C. She, C. Yang, and T. Q. S. Quek. “Radio Resource Management for Ultra-Reliable and Low-Latency Communications”. In: *IEEE Commun. Mag.* 55.6 (June 2017), pp. 72–78 (cit. on p. 5).

[Sim+17] M. Simsek, D. Zhang, D. Öhmann, M. Matthé, and G. Fettweis. “On the Flexibility and Autonomy of 5G Wireless Networks”. In: *IEEE Access* 5 (2017), pp. 22823–22835 (cit. on p. 7).

[SY12] I. Siomina and D. Yuan. “Analysis of Cell Load Coupling for LTE Network Planning and Optimization”. In: *IEEE Transactions on Wireless Communications* 11.6 (June 2012), pp. 2287–2297 (cit. on p. 35).

[Sve11] S. Svensson. *Challenges of Wireless Communication in Industrial Systems*. Keynote. Accessed 2019-07-10. ABB, 2011 (cit. on p. 4).

[TE16] D. Thiele and R. Ernst. “Formal worst-case performance analysis of time-sensitive Ethernet with frame preemption”. In: *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. Sept. 2016, pp. 1–9 (cit. on p. 85).

[Tij95] H. C. Tijms. *Stochastic models, an algorithmic approach*. Reprinted January and April 1995. Chisester: Wiley, 1995 (cit. on p. 83).

[Tri+15] R. Trivisonno, R. Guerzoni, I. Vaishnavi, and D. Soldani. “Towards zero latency Software Defined 5G Networks”. In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. June 2015, pp. 2566–2571 (cit. on p. 107).

[VA07] M. Vlasiou and I. Adan. “Exact solution to a Lindley-type equation on a bounded support”. In: *Operations Research Letters* 35.1 (2007), pp. 105–113 (cit. on p. 79).

[VBK18] H. Volos, T. Bando, and K. Konishi. “Latency Modeling for Mobile Edge Computing Using LTE Measurements”. In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. Aug. 2018, pp. 1–5 (cit. on p. 7).

[Xu+16] Y. Xu, S. E. Elayoubi, E. Altman, R. El-Azouzi, and Y. Yu. “Flow-Level QoE of Video Streaming in Wireless Networks”. In: *IEEE Trans. Mobile Comput.* 15.11 (Nov. 2016), pp. 2762–2780 (cit. on pp. 39, 40, 51, 116, 120, 122).

[Xu+17] Y. Xu, Z. Xiao, H. Feng, T. Yang, B. Hu, and Y. Zhou. “Modeling Buffer Starvations of Video Streaming in Cellular Networks with Large-Scale Measurement of User Behavior”. In: *IEEE Transactions on Mobile Computing* 16.8 (Aug. 2017), pp. 2228–2245 (cit. on pp. 13, 40).

[Xu+13] Y. Xu, S. Elayoubi, E. Altman, and R. El-Azouzi. “Impact of Flow-level Dynamics on QoE of Video Streaming in Wireless Networks”. In: *INFOCOM, 2013 Proc. IEEE*. Apr. 2013, pp. 2715–2723 (cit. on pp. 13, 36, 39–41, 47, 51).

[ZNS14] L. Zhong, K. Nakauchi, and Y. Shoji. “Performance analysis of application-based QoS control in software-defined wireless networks”. In: *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*. Aug. 2014, pp. 464–469 (cit. on pp. 9, 27).

Publications of the Author

Journal and Magazine Publications

[Höß+19a] T. Hößler, L. Scheuvens, **P. Schulz**, A. Noll-Barreto, M. Simsek, and G. Fettweis. "Dynamic Connectivity for Resilient Applications in Rayleigh-Fading Channels". In: *IEEE Communications Letters* (2019). Accepted (cit. on p. 6).

[Höß+20] T. Hößler, **P. Schulz**, E. Jorswieck, M. Simsek, and G. Fettweis. "Stable Matching for Wireless URLLC in Multi-Cellular, Multi-User Systems". In: *IEEE Transactions on Communications* (2020). In major revision (cit. on p. 6).

[Sch+20a] L. Scheuvens, **P. Schulz**, T. Hößler, A. Noll Barreto, and G. Fettweis. "Wireless Control Communications Codesign via State-Aware Resource Allocation". In: *IEEE Communication Letters* (2020). To be submitted (cit. on p. 6).

[Sch+17] **P. Schulz**, M. Matthé, H. Klessig, et al. "Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture". In: *IEEE Communications Magazine* 55.2 (Feb. 2017), pp. 70–78 (cit. on pp. 2, 4, 7).

[Sch+19b] **P. Schulz**, A. Wolf, G. P. Fettweis, et al. "Network Architectures for Demanding 5G Performance Requirements: Tailored Toward Specific Needs of Efficiency and Flexibility". In: *IEEE Vehicular Technology Magazine* 14.2 (June 2019), pp. 33–43 (cit. on pp. 5, 71, 72, 74).

[Sch+20b] **P. Schulz**, H. Klessig, M. Simsek, and G. Fettweis. "Queuing-theoretic Modeling and Improving Buffered Video Streaming QoE in Interference-Limited Cellular Networks". In: *IEEE Transactions on Multimedia* (2020). In major revision (cit. on pp. 10, 39, 54, 60).

[Wol+19] A. Wolf, **P. Schulz**, M. Dörpinghaus, J. C. S. Santos Filho, and G. Fettweis. "How Reliable and Capable is Multi-Connectivity?" In: *IEEE Transactions on Communications* 67.2 (Feb. 2019), pp. 1506–1520 (cit. on p. 6).

Conference Publications

[Höß+19b] T. Hößler, **P. Schulz**, M. Simsek, and G. Fettweis. "Mission Availability for URLLC in Wireless Networks". In: *IEEE Globecom 2019*. 2019 (cit. on p. 6).

[Kle+17] H. Klessig, H. Kuntzschmann, L. Scheuvens, B. Almeroth, **P. Schulz**, and G. Fettweis. “Twitter as a Source for Spatial Traffic Information in Big Data-Enabled Self-Organizing Networks”. In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. San Francisco, Mar. 2017, pp. 1–5 (cit. on p. 17).

[SKF17] **P. Schulz**, H. Klessig, and G. Fettweis. “On Modeling and QoE Evaluation of Buffered Video Streaming in Multi-Cellular Networks”. In: *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. San Francisco, Mar. 2017, pp. 1–6 (cit. on pp. 10, 39, 54).

[Sch+19a] **P. Schulz**, L. Ong, P. Littlewood, B. Abdullah, M. Simsek, and G. Fettweis. “End-to-End Latency Analysis in Wireless Networks with Queuing Models for General Prioritized Traffic”. In: *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. May 2019, pp. 1–6 (cit. on pp. 10, 80, 84).

[Sch+19c] **P. Schulz**, L. Ong, B. Abdullah, M. Simsek, and G. Fettweis. “End-to-End Latency Distribution of Future Mobile Communication Networks”. In: *24th International ITG Workshop on Smart Antennas (WSA 2020)*. Accepted. 2019 (cit. on pp. 10, 27).

[Uzi+18a] S. Uziel, B. Eichhorn, M. Katzschmann, T. Elste, M. Matthé, and **P. Schulz**. “Echtzeit-Antriebsregelung über eine niedriglatente Funkkommunikation”. In: *Jahreskolloquium "Kommunikation in der Automation - KommA*. Lemgo, Nov. 2018.

[Uzi+18b] S. Uziel, T. Elste, B. Eichhorn, M. Katzschmann, and **P. Schulz**. “Low latency wireless closed loop control of an inverted pendulum”. In: *The Conference on Design and Architectures for Signal and Image Processing 2018 (DASIP) - Demo Night*. Porto, Oct. 2018.

[Wol+17] A. Wolf, **P. Schulz**, D. Öhmann, M. Dörpinghaus, and G. Fettweis. “On the Gain of Joint Decoding for Multi-Connectivity”. In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. Singapore, Dec. 2017, pp. 1–6 (cit. on p. 6).

[Wol+18] A. Wolf, **P. Schulz**, D. Öhmann, M. Dörpinghaus, and G. Fettweis. “Rate-reliability tradeoff for multi-connectivity”. In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. Barcelona, Apr. 2018, pp. 1–6 (cit. on p. 6).

Source Code

[Sch19] **P. Schulz**. *Distribution Class*. MATLAB Central File Exchange. 2019. URL: `https://www.mathworks.com/matlabcentral/fileexchange/72423` (visited on Aug. 13, 2019) (cit. on p. 28).