

Ulrich Hofmann

# Internet Modeling with Julia

Models, Algorithms and Programs

 Springer

---

# Internet Modeling with Julia

---

Ulrich Hofmann

# Internet Modeling with Julia

Models, Algorithms and Programs

 Springer

Ulrich Hofmann  
Klingenberg, Germany

ISBN 978-3-658-44691-8      ISBN 978-3-658-44692-5 (eBook)  
<https://doi.org/10.1007/978-3-658-44692-5>

Translation from the German language edition: “Internet-Modellierung mit Julia” by Ulrich Hofmann, © Der/die Herausgeber bzw. der/die Autor(en), exklusiv lizenziert an Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2022. Published by Springer Fachmedien Wiesbaden. All Rights Reserved.

This book is a translation of the original German edition “Internet-Modellierung mit Julia” by Ulrich Hofmann, published by Springer Fachmedien Wiesbaden GmbH in 2022. The translation was done with the help of an artificial intelligence machine translation tool. A subsequent human revision was done primarily in terms of content, so that the book will read stylistically differently from a conventional translation. Springer Nature works continuously to further the development of tools for the production of books and on the related technologies to support the authors.

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Fachmedien Wiesbaden GmbH, part of Springer Nature 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH, part of Springer Nature.

The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

If disposing of this product, please recycle the paper.

---

## Preface

In this book, the Julia programming language is used for selected topics of Internet modeling. The “common thread” is the layered architecture of the Internet: from application to technology. For network operators, beyond the technical performance parameters of transmission rate and transmission duration of IP packets, the user satisfaction values *Quality of Experience* (QoE) with voice, video, haptics transmission are important. With the increase in transmission performance, the old point-to-point lines are also being replaced by packet transmission in the *Internet of Things* (IoT), and the topic of *Quality of Control* (QoC) is gaining importance, currently in the area of assisted/autonomous driving.

The complex processes on the internet elude exact mathematical *White-Box-Model* modeling, and approximate learning modeling methods are being intensively researched and integrated into the standards of network architectures.

In research projects and lectures of the author, Matlab was used and the desire for a license-free laboratory software led via Octave to Julia. The developers from MIT describe their goals as follows [1]:

“We want a language that’s open source, with a liberal license. We want the speed of C with the dynamism of Ruby. We want a language that’s homoiconic, with true macros like Lisp, but with obvious, familiar mathematical notation like Matlab. We want something as usable for general programming as Python, as easy for statistics as R, as natural for string processing as Perl, as powerful for linear algebra as Matlab, as good at gluing programs together as the shell. Something that is dirt simple to learn, yet keeps the most serious hackers happy. We want it interactive and we want it compiled.”

Meanwhile, Julia has evolved into an established programming environment for mathematical models. At annual multi-day JuliaCon conferences, the developers present their new modules. Julia User Groups exist everywhere, including in Munich and Vienna. The advantages of Julia for the evaluation of simulation results have been discovered for internet modeling with simulation tools [2].

The development of Julia is driven by a vibrant ecosystem (<https://julialang.org/>). Beyond a programming environment for the usual operating systems, there is the CUDA

interface for GPU integration and the execution in the JuliaHub Cloud is also offered. We will not further discuss the interesting, intensively conducted discussion on performance comparisons with other programming environments, e.g. [3–6].

This book is intended for a target audience that

- has basic knowledge of internet technology,
- likes working with Matlab and therefore has easy access to Julia,
- is interested in current topics of internet technology.

Each chapter begins with a description of the practical background. This is followed by the development of the model and the implementation with Julia, including the *a-picture-is-worth-a-thousand-words* presentation of results in diagrams. All Julia programs are available for download at <https://github.com/buchjulia/code> and are labeled with the corresponding figure numbers. The most important pieces of code are explained in the book text. Readers who are only interested in new internet topics can skip the Julia sections. Updates and pseudocodes for the programs can be found at [www.qos-analytics.de](http://www.qos-analytics.de).

The author would like to thank M. Bahe, M. Herlich for their helpful comments on Chaps. 6 and 7 as well as the developers of Extremes.jl J. Jalbert, G. Gobeil for their advice on working with the module.

September 2022

Ulrich Hofmann

---

# Contents

<b>1</b>	<b>Introduction</b>	1
1.1	Internet Models	1
1.1.1	Overview	1
1.1.2	Quality of Experience (QoE)	2
1.1.3	Quality of Control (QoC)	3
1.1.4	Load Models	3
1.1.5	Access Control	4
1.1.6	Infrastructure	5
1.1.7	Load Prediction	5
1.2	Julia	6
1.2.1	Getting Started	6
1.2.2	Diagrams, Animations with Plot.jl	7
<b>2</b>	<b>Quality of Experience, QoE</b>	11
2.1	Introduction	11
2.2	Multimedia Applications	12
2.2.1	Audio	12
2.2.2	Video	14
2.2.3	Web	15
2.3	Fairness	16
2.3.1	Variance	16
2.3.2	Fairness	19
2.4	Mapping QoS to QoE	20
2.4.1	Voice Transmission	20
2.4.1.1	E-Model	20
2.4.1.2	R-Value Approximation	23

2.4.2	Streaming Video . . . . .	25
2.4.2.1	QoE Models . . . . .	25
2.4.2.2	Fair Transmission Rate . . . . .	27
2.5	Suggestions for Further Analyses . . . . .	30
<b>3</b>	<b>Quality of Control, QoC</b> . . . . .	<b>31</b>
3.1	Introduction . . . . .	31
3.2	Assisted Driving . . . . .	35
3.2.1	System Description . . . . .	35
3.2.2	Risk Metric . . . . .	40
3.2.2.1	Reliability . . . . .	40
3.2.2.2	Effective Risk Factor and Packet Losses . . . . .	40
3.2.2.3	Age of Information . . . . .	43
3.3	Extreme Value Modelling . . . . .	44
3.3.1	Introduction . . . . .	44
3.3.2	Theorem of Pickands-Balkema-de Haan . . . . .	45
3.3.3	Mean Residual Life Plot (MRL) . . . . .	47
3.3.4	Determination of GPD Parameters . . . . .	47
3.3.5	Return Level Plots (RLP) . . . . .	48
3.3.6	Extreme Value Analysis with Julia . . . . .	49
3.3.6.1	MRL Plot . . . . .	49
3.3.6.2	GPD Parameter Determination . . . . .	50
3.3.6.3	RL-Plot . . . . .	51
3.3.7	Summary . . . . .	53
3.4	Recommendations for Further Analysis . . . . .	53
<b>4</b>	<b>Load Models</b> . . . . .	<b>55</b>
4.1	Introduction . . . . .	55
4.2	Characteristics of Load Models . . . . .	56
4.2.1	Phases, Correlation . . . . .	56
4.2.2	Influence of Variance Parameter . . . . .	59
4.3	Model Development with Julia . . . . .	61
4.3.1	Development Steps . . . . .	61
4.3.2	Correlation Analysis . . . . .	62
4.3.3	Histogram and Distribution Function . . . . .	62
4.3.4	Distribution Test . . . . .	65
4.4	Standardized Load Models . . . . .	67
4.4.1	Gaming . . . . .	67
4.4.2	FTP (File Transfer Protocol) . . . . .	69
4.4.3	Web Browsing . . . . .	71
4.4.4	Video Streaming . . . . .	73
4.4.5	VoIP, Voice over IP . . . . .	74
4.4.6	M2M, Machine to Machine . . . . .	75



---

4.5	Load Generators	75
4.5.1	Basics	75
4.5.2	Random Numbers	77
4.6	Suggestions for Further Analysis	77
<b>5</b>	<b>Access Control</b>	<b>79</b>
5.1	Introduction	79
5.2	Session Admission	83
5.2.1	Types, Models	83
5.2.1.1	Classes	83
5.2.1.2	Full Sharing Model	87
5.2.1.3	Full Split Model	88
5.2.1.4	Russian Doll Model	88
5.2.2	Optimization	91
5.2.2.1	Simulation Model	91
5.2.2.2	Q-Learning	93
5.3	Scheduling	100
5.3.1	Throughput Increase	100
5.3.2	Reduction of the Transmission Duration	103
5.3.3	Current Developments	105
5.4	Recommendations for Further Analysis	106
<b>6</b>	<b>Infrastructure</b>	<b>107</b>
6.1	Introduction	107
6.2	Wireless Networks	108
6.2.1	Basics	108
6.2.2	Grid Algorithm	110
6.2.3	Implementation with Julia	113
6.2.4	5G Scenarios	117
6.3	Fixed Network	118
6.3.1	Kruskal Algorithm	118
6.3.2	Steiner Tree	120
6.3.3	Routing	121
6.4	Suggestions for Further Analysis	124
<b>7</b>	<b>Prediction Models</b>	<b>125</b>
7.1	Introduction	125
7.1.1	Classifications	125
7.1.2	Examples	130
7.2	Prediction Models with Julia Flux	132
7.2.1	Perceptron Basics	132
7.2.2	Perceptron Network Implementation	133

---

7.2.3	Flux .....	139
7.2.4	Experimenting with Real Measurement Data .....	143
7.2.4.1	Description of the Measurement Data .....	143
7.2.4.2	Long-term Forecast in Hourly Intervals .....	145
7.2.5	GPU Usage.....	146
7.3	Recommendations for Further Analysis.....	147
<b>References</b>	.....	<b>149</b>

---

## About the Author

**Ulrich Hofmann** worked on research and development topics in the field of measuring and evaluating transmission qualities on the Internet at Salzburg Research from 1995 to 2022. The first IP-QoS monitoring tool with a web interface was developed and presented at the ICCG in Phoenix in 2000 (<https://www.salzburgresearch.at/publikation/one-way-delay-measurements-with-cm-toolset-3/>). This was followed by invitations to contribute to the EU-DiffServ project AQUILA, at the VDE for the specification of communication requirements for Smart Grid and publications. The EU modeling project INTERMON ([https://www.ercim.eu/publication/Ercim\\_News/enw55/hofmann.html](https://www.ercim.eu/publication/Ercim_News/enw55/hofmann.html)) was coordinated by Ulrich Hofmann.

The transfer of knowledge took place in numerous development projects with the economy and in teaching events with Matlab in the laboratory. This resulted in the textbook *Modeling of Communication Systems*, Manz Verlag 2000, and three chapters in the reference work *Handbook of Communications Engineering*, Hanser Verlag 2008.



## Abstract

Section 1.1 provides an overview of the subject Chapters 2, 3, 4, 5, 6 and 7. It describes the topics and the implementations with Julia. In Sect. 1.2, we explain how to work with the Julia software. Diagrams are required in all subject chapters. Therefore, the plot function is explained in detail in this chapter with an example.

## 1.1 Internet Models

### 1.1.1 Overview

Numerous excellent books on modelling methods in communication technology have been published, tracing the development from the beginning as monographs. The two fundamental mathematical topics are graph theory and queuing theory.

**Graph Theory:** Communication networks are, *per definitionem*, a system of edges (lines) and nodes (routers, end systems). Calculations of the shortest paths, minimal installation costs, and failure probabilities of (redundant) connections are performed using graph theory.

**Queuing Theory:** Each edge (line) in the network graph is a service system for the internet packets to be transmitted. Given a predetermined configuration of the transmission rates of the lines, the waiting time of the packets in the transmission buffer is particularly important for network management. However, a service process with IP packets waiting in the transmission buffer is a complex model of temporal sequences. The first internet models [7] adopted the Erlang telephony model with stochastically independent inter-arrival times and service (telephony) times. This independence of inputs and

outputs generates the so-called Markov property of the state in the transmission buffer as a birth and death process.

This may be true for short messages that fit into a 1500-byte packet. However, as soon as the message is longer, at least two packets are formed (*Packet Train Fragmentation*), whose lengths and arrival intervals are no longer independent of each other. Further dependencies arise from closed processes of packet generation *Closed Queueing Systems*, e.g. in the Transport *Control Protocol* (TCP). The difficulty in mathematically describing the service processes has led to the hope that good approximations of the models are possible with AI methods [8]. This is also referred to as a (*math.*) *Model-Free-Black-Box* method. However, approximations also use a well or less well suited approximation model (linear, polynomial, Fourier series, neural networks) on a *Black Box*, and the concept of model becomes blurred. Current research [9, 10] is investigating the calculation of waiting times on the internet with a two-stage AI model. In the first stage, the routing topology for all packet flows is mapped: “In which routers do the flows compete?”, in *Recurrent Neuronal Networks*(RNN) for each *Packet Flow* and router. After a few iterative runs, the order of the routers traversed by the *Packet Flow* is stored in the state of the RNN. A *Feed Forward Network* (FFN) is then trained on this, which maps the relationship between the RNN (and thus the routing) and the QoS parameters of the *Packet Flows* such as packet losses and transmission duration. The goal is that the RNN-FFN model, after being trained with many routing variants, is able to approximate the waiting times and packet losses well for routing topologies not yet seen. After answering still open questions [10], this could become an important topic for subsequent books.

### 1.1.2 Quality of Experience (QoE)

The subsequent chapters begin at the highest level of the value chain. Users expect good application quality and are less interested in internet technology. Therefore, for internet technologists, the task is to model the relationship between the quality of packet transmission and application quality. The *Quality-of-Experience*-(QoE)-metric is the required quantification of the evaluation “perception” for audio, video, and haptics.

Once this quantification is defined (e.g., for IP telephony *Mean Opinion Score* (MOS) in the interval [1, 5]), models are needed that allow mapping the IP packet transmission quality *Quality of Service* (QoS) to the QoE value. Only with these mapping formulas  $QoE(QoS)$  can the network technologist configure the network with a QoS so that the required QoE value is achieved. The adjustment of the QoE across all users is the *QoE-Fairness*. Depending on the application (e.g., pixel resolution), the same QoE value can be achieved with different QoS values (transmission rates, packet loss rates). For the explanation of the approximate modelling, the already longer established QoE model for *Voice over IP* (VoIP) is used.

Julia Programs

E-Model: The mapping formula  $QoE(QoS)$  according to the ITU model was derived from psychoacoustic models of the effect of transmission duration and packet losses and is thus a *White-Box* model. A neural network can learn the  $QoE(QoS)$  relationship as an approximation on the *Black Box* with measurements.

QoE-Fairness: The calculation of the QoS values  $QoS_i$  assigned to the end systems  $i$  for a *Fairness* with equal QoE values for all end systems  $QoE_i = QoE_j \forall i, j$  involves solving a nonlinear system of equations.

### 1.1.3 Quality of Control (QoC)

With the increase in the performance of transmission lines (from copper to glass), the aforementioned waiting times of packets became so small that real-time applications with packet transmission became possible. The replacement of the many modem lines between controller and system by a single *Multiaccess* packet network is almost complete (e.g., in the energy sector IEC 61850 standards). Currently, packet transmission for mobile real-time applications is being implemented. The *Car-to-car-/Infrastructure-(C2C/I-)*communication has stringent requirements. The “QoE” would here be interpreted as damage to the road user and QoC (*Quality of Control*) is described with the concept of risk as the probability of occurrence for personal injury. Here too, a mapping formula Risk (QoS) is needed. A pragmatic approach is not to try to find an accurate model across the entire spectrum of situations, but to focus on good models for the risk-relevant extreme situations of exceeding permissible QoS values.

Julia Programs

The extreme value analysis of buffer occupancies is carried out according to the extreme value theorem of Pickands-Balkema-de Haan for threshold exceedances. The MRLP (*Mean Residual Life Plot*) is used to test for fulfillment of the prerequisites. The risk assessment for exceeding critical thresholds is done with the analysis of the RLP (*Return Level Period*).

### 1.1.4 Load Models

As already explained above, the QoS value mainly results from the waiting times in the routers. These packet jams depend (analogous to intersections in road traffic) on the intensity of the packet streams and the transmission rate of the lines. Therefore, models of the packet streams generated by the applications and competing for the transmission capacity are needed. The most important parameters are the packet lengths and

the inter-arrival times. These load models are application-related and are derived from measurements. Initial system designs with these models can then be evaluated in real operation and, if necessary, readjusted. The international standardization organization of network operators, ITU (*International Telecommunication Union*), continuously updates these models. With these load generators, standardized performance tests can also be carried out. Only then are comparisons between products possible.

Julia Programs

For the derivation of the models, distribution tests are carried out: *Quantile-quantile*- (QQ-)diagram and Chi-square test. Once a load model is available, the load generators must be implemented for simulations and laboratory tests.

### 1.1.5 Access Control

The simultaneous access of numerous users to the internet presents both a social and technological issue. Broad discussions about net neutrality and fairness are ongoing. For the social aspect, we direct the reader to the relevant literature and address the two technical levels of access control:

*Access Control* refers to the admission of a communication request. Depending on whether Quality of Service (QoS) values have been agreed upon (in the 5G network 5QI 5G *Quality Identifier*) or if one is satisfied with "somehow" *Best Effort*, the network operator then makes the decision to *Accept* or *Reject* the *Session* after evaluating their remaining transmission capacities. A division of the transmission capacity into *Admission Classes* can also be done based on user groups, or *Tenants*.

*Scheduling* for the pickup/sending of IP packets: After admission, the packets must be transmitted (5G: DL *Downlink*, UL *Uplink*). The *Scheduling* determines the order of packet transmission among the competing *Sessions*.

Julia Programs

The admission probabilities of *Sessions* of the *Admission Classes* are analytically treated for three variants of the division of the existing total transmission rate on a Markov model: *Full Sharing*, *Full Split*, *Russian Doll*. To deviate from the Markov assumption, a simulation is implemented. The search for the optimum is done with *Q-learning*. The *Minimum-bit-rate*- (MBR-)Scheduling is simulated.

### 1.1.6 Infrastructure

The first task is to install a suitable infrastructure of access points, or *Base Stations*, for the end systems in the network. For wireless networks like 5G, this means distributing the *Base Stations* across a territory to achieve minimal costs with maximum radio coverage. The models distinguish between:

- Macro cells with high range ( $> 10$  km), low transmission frequency ( $< 1$  GHz), low bitrate and high transmission power ( $> 10$  W)
- Small/Micro Cells with less than 100 m range, low transmission power and high bitrate.

For the underlying router fixed network, which connects the *Base Stations* among each other and to the outside with the global internet, an optimal topology is calculated with given boundary conditions to the line costs. This is followed by the selection of transmission paths with the *Routing*.

Julia Programs

The grid method is implemented for the positioning of the *Base Stations*. The iterative optimization steps are visualized with an animation.

The fixed network optimization is done for *Minimal Spanning Trees* (MST) with the Kruskal algorithm, the *Steiner-Tree*-model is explained. The *Shortest Path First Routing* (SPF) is determined with the Dijkstra algorithm.

### 1.1.7 Load Prediction

The topic of load prediction is important for many areas of the internet. From this, for example, demand-dependent configuration changes can be derived. Network management can also redefine the routing. Strong deviations between prediction and measurements are also cases of suspicion for load attacks. In all load predictions, the task is to recognize a pattern for the load process in a time scale  $\Delta t$ . *Multi-scale* patterns with day, hour can also be discovered. The prediction formula is determined from the measurements of the past. For the nonlinear models of load prediction, neural networks are again suitable. It is easier to derive the *Black-Box* model from measurements of the temporal correlations of load shifts between radio networks with neural networks than to construct a complicated *White-Box* model with mobility density, travel routes, travel speeds, traffic jam probabilities, etc.