Load Modelling and Generation in IP-based Networks

Andrey Kolesnikov

Load Modelling and Generation in IP-based Networks

A Unified Approach and Tool Support

With a foreword by Prof. Dr. B. E. Wolfinger

Springer Vieweg

Andrey Kolesnikov
Hamburg, Germany

Dissertation with the aim of achieving a doctoral degree at the Faculty of Mathematics, Informatics and Natural Sciences, Department of Computer Science, University of Hamburg, 2017.

ISBN 978-3-658-19101-6 ISBN 978-3-658-19102-3 (eBook)
DOI 10.1007/978-3-658-19102-3

Library of Congress Control Number: 2017951224

Springer Vieweg

© Springer Fachmedien Wiesbaden GmbH 2017
This work is subject to copyright. All rights are reserved 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, express 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.

Printed on acid-free paper

This Springer Vieweg imprint is published by Springer Nature
The registered company is Springer Fachmedien Wiesbaden GmbH
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Foreword

The number of users demanding computer network services still keeps to be increasing at an enormous rate. This is true, in particular, for mobile networks and for the Internet. Moreover, the traffic generated per user is growing significantly so that a study by CISCO estimated the global annual amount of IP traffic in the Internet to be more than 1 ZB ($= 10^{21}$ Byte) by the end of 2016. Also, the data rates required by the data streams to be transmitted tend to become increasingly challenging because, e.g., video communication gets more and more popular accompanied with a strong increase of the video quality demanded by the users. Of course, also the performance of network components will continue to increase at an astonishingly high rate such as in case of computer hardware (multi-core processors), of switching nodes (optical switches) and in particular in data transmission media and techniques (as well in optical as in radio transmissions). Nevertheless, the extremely strong growth in the traffic to be transmitted can be expected to lead to numerous bottlenecks even in the high-speed networks which are currently emerging.

Therefore, performance evaluation of communication systems and computer networks will certainly not become superfluous but, on the contrary, methods and tools will be needed which allow one to analyse (e.g., by means of measurements) how computer networks react to well-defined background loads or traffic peaks. To generate such (artificial but sufficiently realistic) background loads or traffic, dedicated load generators are indispensable, which should be applicable in a flexible and rather general manner. The elaboration of such a broadly applicable load (or traffic) generator, called UniLoG, has been the goal underlying the research documented by this book.

In order to be able to generate highly realistic traffic in computer networks the UniLoG approach is as follows: The load generator produces sequences of requests (representing the load), which are handed over at a service interface within a computer network in the same manner as they would be handed over by a real service user at this interface to the component providing the service. As a consequence of the execution of the requests, communication by means of data units (e.g., video frames, TCP segments, IP packets or Ethernet frames) is initiated, which finally represent the traffic in the network.

The major contributions in this publication – representing the PhD thesis of the author – are impressive. The results achieved comprise:

- a formal description technique (LoadSpec), which allows one to describe load in a unified manner based on a sequence of abstract requests being independent on the concrete interface underlying the load generation,
- a variety of load models (e.g. for voice, video and Web traffic), which are specified by means of the author's formal description technique and considerable effort is spent for a realistic parameterization of all models elaborated,
- design of a highly modular architecture for the UniLoG load generator and full implementation of this tool in a very efficient manner,
- accomplishment of a geographically distributed version of UniLoG, based on the manager-agent-paradigm,
- realization of various adapters for very different interfaces of a computer network such as service interfaces of IPv4, TCP, UDP and HTTP, which proves successfully a unique feature of UniLoG, namely being usable to generate load at all interfaces of a complete network protocol stack (besides Physical Layer),
- various case studies which, e.g., demonstrate that UniLoG can indeed be used to generate highly complex background loads in a very realistic manner.

This innovative research report not only contains a lot of conceptually and theoretically interesting ideas but the results are also practically relevant. Among others they should be a valuable source of information for Internet Service Providers (ISPs) or Telecoms who are responsible for providing efficient network services in large and complex networks. Moreover, the results achieved should also be of significant relevance to researchers, developers of new network and distributed application services, network administrators, etc., who might be interested, e.g., to analyse what impact new services, change of user behaviour, or increasing load could have on network performance and user's quality of experience (QoE).

Hamburg, in June 2017 *Bernd E. Wolfinger*

Preface

The accurate and realistic modelling and generation of network workload which may consist of a mix of many complex traffic sources is a difficult and challenging task. Analyses and generation of network workload, in particular in large-scale networks, can be aggravated by the heterogeneity and large number of used network devices and protocols, as well as different types of applications and services which may strongly evolve over time. Furthermore, the purpose of the workload modelling and, therefore, the objectives of the corresponding experimental tests and case studies may vary, e.g., from the performance evaluation analyses to the analyses of network neutrality and security mechanisms. Therefore, in order to keep up with the perpetually emerging new requirements and the corresponding technical challenges, networking research community needs to continuously improve the methods and tools used for workload modelling and generation.

In this thesis, a unified approach for workload modelling and generation with general applicability in IP-based networks is elaborated and a set of the corresponding tools for the specification and generation of synthetic workloads is developed. The architecture of the Unified Load Generator `UniLoG` proposed and implemented in the thesis can be used for the generation of realistic workloads and traffic according to various workload and traffic models at different (e.g, application, transport, and network) service interfaces in IP-based networks. The proposed `UniLoG` architecture provides a high degree of flexibility, extensibility, and scalability in the workload generation process. Further, a set of concrete workload models for exemplarily chosen types of traffic sources (such as VoIP, video, and Web traffic) is elaborated and provided for load generation with `UniLoG`. Several experimental results related to the study of “hot topics” like performance and QoS analysis of video streaming applications are presented and emphasize how the proposed `UniLoG` load generator advances the state-of-the-art in workload modelling and generation.

Contents

I Introduction and Fundamentals 1

1 Introduction 3

1.1 Motivation 3

1.2 Objectives and Scope of the Thesis 8

1.3 Structure of the Thesis 10

2 Foundations and Research Field 17

2.1 Workload Modelling and Specification Techniques 18

2.1.1 Selected Workload Modelling Techniques 23

2.1.2 Selected Workload Specification Techniques 27

2.2 State-of-the-Art in Workload Generation 30

2.2.1 Web Workload and Traffic Generation 34

2.2.2 Traffic Generation at Transport Layer Service Interfaces 40

2.2.3 Traffic Generation at Network Layer Service Interfaces 44

2.2.4 Traffic Generation at Data Link Layer Service Interfaces 47

2.2.5 Workload Tests in Research and Industry 55

2.3 A Unified Approach to Workload Modelling and Generation in Computer Networks 56

II Workload Specification and Modelling 61

3 A Formal Workload Description Technique 63

3.1 The Basic Concept of User Behaviour Automata 63

3.2 Generalisation of the Basic Concept of User Behaviour Automata 68

3.2.1 Definition of the Generalised User Behavior Automaton (UBA) 69

3.2.2 Specification of Transitions between Elementary States 73

3.2.3 Aggregation of User States into Macro-States 76

3.3 XML Schema Definition for the UBA Components 77
3.3.1 XSD Simple Elements 79
3.3.2 XSD Complex Elements 81
3.3.3 Introduction to the UBA Schema 81
3.4 Description of Abstract Requests and System Reactions . . . 84
3.4.1 Relevant Abstract Request Types 85
3.4.2 Semantics of Abstract Request Types 86
3.4.3 Definition of Abstract Request Types in the UBA Schema 89
3.5 Specification of Values for UBA Parameters 91
3.6 Syntax Rules for Context Expressions 94
3.7 Specification of Complex User Environments 95

4 Examples of Load Models for Different Traffic Sources 99
4.1 Models for Speech Traffic Sources 99
4.1.1 Voice Codecs with Constant Bit Rate 100
4.1.2 Voice Codecs with Silence Detection 102
4.2 Modelling of Video Traffic Sources 108
4.2.1 Modelling of the GOP Structure 109
4.2.2 Statistical Characterization and Modelling of Frame Lengths 113
4.2.3 Partitioning into Shot Classes 126
4.3 Modelling of Web Workloads with UniLoG 131
4.3.1 UniLoG Approach for Web Workload Modelling and Generation 134

III Workload Generation 143

5 Architecture of the Unified Load Generator 145
5.1 Basic Requirements 145
5.1.1 Functional Requirements 145
5.1.2 Non-functional Requirements 146
5.2 Overview of the UniLoG Architecture 150
5.3 Generator Functionality 154
5.4 Adapter Functionality 155
5.5 Real-time Requirements of Requests 158
5.5.1 Impact of Multitasking 158
5.5.2 Latency Introduced by System Calls 160
5.5.3 Latency Introduced by UniLoG Components 162

5.5.4 Intrinsic Model Factors 163
5.6 UniLoG Design Extensions for Multi-Core Platforms 163

6 Distributed UniLoG Architecture 167
6.1 Prerequisites and Requirements 167
6.2 System Architecture for Distributed Load Generation 172
6.3 Implementation Aspects 175

7 Load Generation at Network Layer Service Interfaces 177
7.1 Application Scenarios and Requirements 177
7.2 Design of the UniLoG.IPv4 Adapter 179
7.2.1 Abstract IP Requests 182
7.2.2 Types of Abstract System Reactions 186
7.3 Performance Evaluation 188

8 Load Generation at Transport Layer Service Interfaces 195
8.1 Design and Architecture of the UniLoG.TCP Adapter 196
8.1.1 Supported Types of Abstract Requests 196
8.1.2 TCP Load Receivers 202
8.1.3 Supported Types of Abstract System Reactions 203
8.1.4 Types of Supported Traffic Matrices 205
8.2 Performance Evaluation 205
8.3 Aspects of the UniLoG.UDP Adapter Implementation 215

9 Generation of Web Workloads 217
9.1 Architecture of the UniLoG.HTTP Adapter 217
9.2 Implementation Aspects 220
9.2.1 Browser Imitation 222
9.2.2 Browser Integration 224
9.3 Construction of the Pool of Web Sites 227
9.3.1 Estimation of Abstract Workload Characteristics . . . 229
9.3.2 Measurement Results 240

IV Applications of the UniLoG Load Generator 249

10 Estimation of QoS Parameters for RTP/UDP Video Streaming 251
10.1 Experimental Network 252
10.2 Configuration of the Background Load 255
10.3 Streaming Quality Metrics 257
10.4 Results and Discussion 258

11 Estimation of QoS Parameters for RTP/TCP Video Streaming 263
11.1 Experimental Network 265
11.2 Configuration of the TCP Background Load 268
11.3 Measurement Results and Discussion 269
11.3.1 Streaming in the Experimental Network without Background Load 270
11.3.2 Streaming in the Experimental Network under Background Load 273
11.4 Conclusions 278

V Results and Conclusions 281

12 Summary and Outlook 283
12.1 Summary of Results 283
12.2 Outlook on Future Work 286

A Context Expression Functions 287

Bibliography 291

List of Acronyms

ACPI	Advanced Control and Power Interface
AJAX	Asynchronous JavaScript and XML
AMR	Adaptive Multi-Rate
API	Application Programming Interface
ATM	Asynchronous Transfer Mode
AQM	Active Queue Management
BBB	Big Buck Bunny
BMAP	Batch Markovian Arrival Process
BRAS	Broadband Remote Access Server
CA	Certification Authority
CBR	Constant Bit Rate
CCP	Compression Control Protocol
CDF	Cumulative Distribution Function
CDN	Content Delivery Network
CET	Central European Time
CIFS	Common Internet File System
COM	Component Object Model
COTS	Commercial off-the-shelf
CPU	Central Processing Unit
CSRF	Cross-Site-Request-Forgery
CSS	Cascading Style Sheet
DCCP	Datagram Congestion Control Protocol
DNS	Domain Name System
DOM	Document Object Model
DPDK	Data Plane Development Kit
DSCP	Differentiated Service Code Point

DSL	Digital Subscriber Line
ECDF	Empirical Cumulative Distribution Function
ECN	Explicit Congestion Notification
EPMF	Empirical Probability Mass Function
EQ	Event Queue
FCFS	First-Come, First-Served
FDT	Formal Description Technique
FEC	Forward Error Correction
FSM	Finite State Machine
FTP	File Transfer Protocol
GOP	Group of Pictures
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HDTV	High Definition Television
HMM	Hidden Markov Model
HPET	High Precision Event Timer
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IANA	Internet Assigned Numbers Authority
IAT	Inter-arrival Time
ICMP	Internet Control Message Protocol
IDPS	Intrusion Detection and Prevention System
iLBC	Internet Low Bit Rate Codec
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IPTV	Internet Protocol Television
iSAC	internet Speech Audio Codec
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider

ITU	International Telecommunication Union
JSON	JavaScript Object Notation
LAN	Local Area Network
MAC	Medium Access Control
MIME	Multi-Purpose Internet Mail Extensions
MLE	Maximum Likelihood Estimator
MMPP	Markov Modulated Poisson Process
MPEG	Moving Picture Experts Group
MPI	Message Passing Interface
MTU	Maximum Transmission Unit
NALU	Network Abstraction Layer Unit
NAT	Network Address Translation
NIC	Network Interface Card
NNTP	Network News Transfer Protocol
NPT	Network Port Translation
NTP	Network Time Protocol
OS	Operating System
OSI	Open Systems Interconnection
PCAP	Packet Capture
PCM	Pulse Code Modulation
PDF	Probability Density Function
PDU	Protocol Data Unit
POP3	Post Office Protocol Version 3
PPBP	Poisson Pareto Burst Process
PPP	Point-to-Point Protocol
PTP	Precision Time Protocol
QoS	Quality of Service
QPC	QueryPerformanceCounter
RQ	Request Queue
RMI	Remote Method Invocation
RTCP	RTP Control Protocol

RTP	Real-Time Transport Protocol
RTSP	Real-Time Streaming Protocol
SAP	Service Access Point
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SDU	Service Data Unit
SIP	Session Initiation Protocol
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SNTP	Simple Network Time Protocol
SOA	Service-oriented Architecture
SUT	System Under Test
TCP	Transmission Control Protocol
TLS	Transport Layer Security
ToS	Type of Service
TSC	Time Stamp Counter
TTL	Time to Live
UBA	User Behavior Automaton
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VAD	Voice Activity Detection
VBR	Variable Bit Rate
VoD	Video on Demand
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WLAN	Wireless Local Area Network
WWW	World Wide Web
XML	Extensible Markup Language
XSD	XML Schema Definition

List of Figures

2.1 Unified approach to workload modelling illustrated for the case of modelling at the IPv4 network service interface. 58

3.1 A UBA on the level of macro-states. 64
3.2 Modelling of a user at the Web browser command interface by means of a UBA. 66
3.3 Modelling of a user at the HTTP service interface by means of a UBA. 67
3.4 The classification of data types supported by XML Schema. . . . 80
3.5 UBA schema definition file. 82
3.6 An example of a UBA model file including a reference to the UBA schema. 84
3.7 Mapping of the abstract request types from the modelling domain onto the system calls at the TCP socket interface. 88
3.8 Specification of valid abstract request types in the schema. . . . 90
3.9 Definition of the request type `SendDataBlock` to model the TCP socket `send()` call. 91
3.10 Definition of the complex type `ValueSpec` for the specification of values for different UBA parameters. 92

4.1 A UBA model of the G.711 output voice stream, control information ignored. 102
4.2 A UBA model of the G.711 output voice stream, control information included. 103
4.3 Output stream of the G.723.1 codec with silence intervals and talkspurts interrupted by short breaks. 104
4.4 A UBA describing an ON/OFF model for the voice stream from G.723.1 codec. 105
4.5 A simple UBA model using a chain of six R-states to represent the *IBBPBB*... sequence of H.264-coded video frames. 110

4.6 A universal UBA model using three R-states and conditional state transitions to model different possible GOP structures. . . 111
4.7 IP throughput of the BBB video stream at the client side (RTSP streaming using RTP over UDP in a 100 Mbit/s Fast Ethernet). 114
4.8 I-frame lengths (in bytes) from BBB video: EPMF, fit by means of the Log-normal and Gamma PDFs. 115
4.9 I-frame lengths (in bytes) from BBB video: ECDF, fit by means of the Log-normal and Gamma CDFs. 116
4.10 P-frame lengths (in bytes) from BBB video: EPMF, fit by means of the Log-normal and Gamma PDFs. 120
4.11 P-frame lengths (in bytes) from BBB video: ECDF, fit by means of the Log-normal and Gamma CDFs. 121
4.12 B-frame lengths (in bytes) from BBB video: EPMF, fit by means of the Log-normal and Gamma PDFs. 123
4.13 B-frame lengths (in bytes) from BBB video: ECDF, fit by means of the Log-normal and Gamma CDFs. 124
4.14 Partitioning of the GOPs into shot classes using geometric boundaries. 127
4.15 A universal UBA model for H.264-coded video sources with two shot classes. 129
4.16 Modelling approach MA_1. 132
4.17 Modelling approach MA_2. 132
4.18 Modelling approach MA_3. 133
4.19 Retrieval of multiple pages from the Web server `www.foo.com` and the corresponding UBA model. 134

5.1 Overview of the basic `UniLoG` architecture. 150

6.1 Architecture of the system for distributed load generation on the basis of the `UniLoG` load generator. 173

7.1 Architecture of the `UniLoG.IPv4` adapter. 179
7.2 Example of the abstract request type `InjectIPPacket` to model the generation of IPv4 packets at the network service interface. . 183
7.3 Experiment duration (time to generate $10 \cdot 10^6$ IPv4 `send()` requests) for different IPv4 payload sizes. 190
7.4 The rate of blocking IPv4 `send()` requests achievable for different IPv4 payload sizes. 191
7.5 The data rate of the IPv4 flow achievable on a Gigabit Ethernet link for different IPv4 payload sizes. 191

8.1 Architecture and basic components of the `UniLoG.TCP` adapter. . 197
8.2 Definition of the abstract TCP request type `TCPOpenRequest (localTCPPort, remoteIPAddress, remoteTCPPort)` to model the generation of active TCP connection requests. 198
8.3 Definition of the abstract TCP request type `TCPSendRequest (payloadBuffer, payloadLength)` to model the transmission of user data from the `payloadBuffer` array. 199
8.4 Definition of the abstract TCP request type `TCPCloseRequest (localTCPPort, remoteIPAddress, remoteTCPPort)` to model the connection close requests. 201
8.5 Experiment duration (the time required to generate a total of $10 \cdot 10^6$ TCP `send()` requests) for different TCP payload sizes. . 209
8.6 The rate of blocking TCP `send()` requests achievable for different TCP payload sizes. 210
8.7 The data rate of the TCP stream achievable on a Gigabit Ethernet link for different TCP payload sizes. 210

9.1 Architecture and basic components of the `UniLoG.HTTP` adapter. 218
9.2 Algorithm used to find the best matching HTTP request from the pool. 219
9.3 Modules involved into the analysis of induced HTTP traffic and estimation of the abstract Web workload characteristics. 232
9.4 List of the connections established to the server(s) which are involved into the delivery of the page `www.example.com`. 238

10.1 Experimental network: 100 Mbit/s Fast Ethernet, 54 Mbit/s IEEE 802.11g WLAN, transmission of the BBB video stream by means of RTP over UDP. 253
10.2 Quality metrics for the BBB video stream observed under different levels of background loads. 260

11.1 Experimental network: 1 Gbit/s Gigabit Ethernet, 54 Mbit/s IEEE 802.11g WLAN, transmission of the BBB video stream by means of RTP over TCP. 266
11.2 IP throughput and streaming statistics of the BBB video stream (no background load, TCP receive buffer size `RCVBUFF` in the VoD client set to 17520 Byte). 271
11.3 IP throughput and streaming statistics of the BBB video stream (no background load, TCP receive buffer size `RCVBUFF` in the VoD client increased to 65535 Byte). 272

11.4 IP throughput of the BBB video stream. 274
11.5 Jitter values in the BBB video stream. 275
11.6 Number of RTP sequence errors in the BBB video stream. 276
11.7 Number of lost RTP packets in the BBB video stream. 277
11.8 Duplicate TCP segments in the BBB video stream. 278

List of Tables

4.1 Packet Types in the G.711 and the G.729.1 Codecs 100

4.2 Measurements of *ON*/*OFF* phase durations (classical approach and approach from [MBM09]) using the set of typical telephone conversations available in [BAS96]. 107

4.3 Geometric partitioning of the BBB video into n shot classes for $n = 2, 3, \ldots, 7$. 128

4.4 Inter-shot class transition probability matrix P for $n = 7$ shot classes S_1–S_7. 130

6.1 Control commands supported by the `UniLoG` load agents. . . 174

8.1 TCP `send()` request rate and utilization of the Gigabit Ethernet link measured during the generation of $10 \cdot 10^6$ requests using different sizes of the TCP send buffer. 208

9.1 Summary of measurement results for abstract Web workload characteristics of Web pages. 242

9.2 Number of pages in different classes of `inducedServerLoad` and `inducedServerProcessingTime`. 244

10.1 Parameters of the IP background load streams in the case study. 256

11.1 Parameters of the TCP background load streams. 269