
Dissertation

**Modell-basierte Leistungsbewertung und
Optimierung von Multi-Core-Architekturen zur
Paketverarbeitung in Kommunikationsnetzen**

zur Erlangung des akademischen Grades

Dr. rer. nat.

an der Fakultät für Mathematik, Informatik und

Naturwissenschaften der Universität Hamburg

eingereicht beim Fachbereich Informatik von

Torsten M. Runge

Hamburg, 2015

- | | |
|---------------|------------------------------|
| 1. Gutachter: | Prof. Dr. Bernd E. Wolfinger |
| 2. Gutachter: | Prof. Dr. Thomas Ludwig |
| 3. Gutachter: | Prof. Dr. Otto Spaniol |

Einreichung der Dissertationsschrift: 7. Dezember 2015

Verteidigung der Dissertationsschrift: 2. Mai 2016

Berichte aus dem Forschungsschwerpunkt
Telekommunikation und Rechnernetze

Band 10

Torsten M. Runge

**Modell-basierte Leistungsbewertung und
Optimierung von Multi-Core-Architekturen zur
Paketverarbeitung in Kommunikationsnetzen**

Shaker Verlag
Aachen 2016

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Hamburg, Univ., Diss., 2016

Copyright Shaker Verlag 2016

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-4587-1

ISSN 1439-3573

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • E-Mail: info@shaker.de

Vorwort des Herausgebers

Wie sämtliche bisherigen publizierten Bände der vorliegenden Schriftenreihe, so ist auch die hier veröffentlichte Arbeit als Dissertationsschrift beim Fachbereich Informatik der Universität Hamburg eingereicht und erfolgreich verteidigt worden. Gleichzeitig ist es der zehnte publizierte Band und daher de facto ein kleiner Jubiläumsband.

Durch die immensen Fortschritte in der Verarbeitungsleistung heutiger PC-Systeme (zum einen durch deutlich schnellere Hardware sowie zum anderen durch Einsatz von Multi-Core-Systemen) wird es zunehmend attraktiv, auch Rechnernetzkomponenten nicht mehr nur unter Verwendung von Spezial-Hardware zu realisieren. Das gilt insbesondere für Hardware-Router in IP-basierten Rechnernetzen, die sich – zwar mit einem evtl. deutlich reduzierten Paketdurchsatz, jedoch mit einem beträchtlichen Gewinn an Flexibilität (Programmierbarkeit) – durch Software-Router ersetzen lassen. Daher stellt die Untersuchung der Eignung aktueller Multiprozessor-PC-Systeme zur Realisierung von Software- Routern in Kommunikationsnetzen mit schneller Datenübertragung (z.B. in Gigabit- und 10-Gigabit-Netzen) eine interessante und sehr praxisrelevante Forschungsaufgabe dar. Insbesondere müssen dabei Treiber, Netzwerkprotokollstapel und Anwendungen möglichst gut an die Hardware-Architektur und deren Parallelisierungsmöglichkeiten des eingesetzten PC-Systems angepasst werden.

Das im Mai 2015 erfolgreich abgeschlossene MEMPHIS-Projekt befasste sich als ein von der DFG gefördertes Kooperationsprojekt des Lehrstuhls für Netzarchitekturen und Netzdienste (Prof. Carle) an der TU München und des Arbeitsbereichs Telekommunikation und Rechnernetze (Prof. Wolfinger) an der Universität Hamburg mit der Leistungsbewertung und Verbesserung der Paket- und Protokollverarbeitung auf aktuellen Multiprozessor-PC-Systemen.

Die Dissertation von Herrn Runge sollte nun die folgenden Beiträge liefern, die gleichzeitig auch zentrale Beiträge zu dem DFG-Projekt MEMPHIS darstellen: Zum einen sollte die Paketverarbeitung in einem Linux-Software-Router gründlich analysiert werden und ausgehend von diesem tieferen Verständnis ein detailliertes

Router-Modell in den Netzsimulator ns-3 integriert werden. Dabei stellte sich heraus, dass die vorliegende Version von ns-3 (ähnlich wie andere Simulationswerkzeuge für Rechnernetze bzw. Netzkomponenten) keine angemessene Berücksichtigung der rechnerinternen Betriebsmittelverwaltung gestattete. Dies nahm Herr Runge dann zum Anlass – als einer der Hauptbeiträge seines Dissertationsprojektes – ein sehr allgemeines Konzept für die Modellierung der rechnerinternen Ressourcenverwaltung zu erarbeiten und exemplarisch in ns-3 zu integrieren. Erst dadurch konnte ein hinreichend realistisches Router-Modell erzeugt werden. Das resultierende Simulationsmodell sollte dann anhand von Messungen (seitens des MEMPHIS-Projektpartners an der TU München) kalibriert und anschließend validiert werden. Sofern möglich sollte schließlich auch, durch Veränderung der ursprünglichen Paketverarbeitung, versucht werden, die Leistung des ursprünglich betrachteten Software-Routers signifikant zu verbessern.

Im Einzelnen beinhaltet diese Arbeit die folgenden zentralen Resultate:

- Nach Einführung von Begriffsdefinitionen für Leistungskenngrößen, die für seine Dissertation von zentraler Bedeutung sind, fasst Torsten Runge im zweiten Kapitel zunächst die bekannten Gesetze von Moore und Amdahl zusammen. Es folgt eine gut nachvollziehbare Klassifikation von Rechnerarchitekturen mit Fokussierung auf Rechner, die gezielt Parallelverarbeitung unterstützen. Der Hauptbeitrag von Kapitel 2 kann indes gesehen werden in der sehr sorgfältigen und kompetenten Analyse des State-of-the-Art für die Bereiche Paketverarbeitung auf Standard-Hardware (z.B. durch Software-Router) sowie deren Modellierung (mit entweder analytischer oder simulativer Modellauswertung).
- Kapitel 3 gibt einen interessanten und nützlichen Überblick über moderne Rechnerarchitekturen mit Fokus auf Multi-Core-Architekturen. Grundprinzipien dieser Rechnerarchitekturen werden in einer sehr kompetenten Art und Weise erläutert und sind dadurch für den Leser gut nachvollziehbar. Stets wird auch die zugrundeliegende Motivation für den Einsatz der entsprechenden Prinzipien gegeben, die in der Regel aus dem Wunsch nach Verbesserung der Leistungsfähigkeit der Rechnerarchitektur resultiert.

-
- Das Kapitel 4 widmet sich gezielt Innovationen bei Betriebssystemen (insbesondere Linux) und speziell auch bei Softwarekomponenten für die Paketverarbeitung in Rechnern mit Standard-Hardware. Wie bereits in dem Hardware-orientierten dritten Kapitel, so ist auch Kapitel 4 sehr breit angelegt und deckt eine Vielzahl von Mechanismen ab, die heutzutage zur Verfügung stehen, um signifikante Effizienzsteigerungen bei der Paketverarbeitung in PCs mit Multi-Core-Architektur zu erzielen.
 - Kapitel 5 kann als zentrales Herzstück der vorliegenden Dissertation angesehen werden. Hier wird ein Ansatz zur Modellierung rechnerinterner Betriebsmittelverwaltung entwickelt und dann auch präzise spezifiziert, der sich durch eine sehr allgemeine Anwendbarkeit auszeichnet. Der innovative Modellierungsansatz berücksichtigt eine umfangreiche Liste von Anforderungen (funktionaler und nicht-funktionaler Natur). Zentrale Bausteine des neuartigen Ansatzes sind die *Task Units* (zur Modellierung von Datenverarbeitungsschritten bei der rechnerinternen Bearbeitung von Aufträgen), die *Resources* (zur Modellierung der Betriebsmittel, die durch Task Units bei deren Ausführung benötigt werden) sowie die *Resource Manager* (zur Modellierung der Betriebsmittelvergabe an die Task Units). Dabei kommen die Resource Manager (RM) in zwei Ausprägungen vor: zum einen die *Local RMs* sowie zum anderen der *Global RM*. Auch die grundsätzliche Entscheidung, die Modellbausteine Task Unit, Resource und Resource Manager in drei Ebenen/Schichten (Processing Plane, Resource Plane und Resource Management Plane) zu organisieren, stellt ein ausgezeichnetes Hilfsmittel für den Modellierer dar, in dessen Bestreben, übersichtliche und gut verständliche Modelle zu erzeugen. Das Verhalten sämtlicher zentraler Modellierungsbausteine wird im Übrigen mittels UML-Zustandsdiagrammen spezifiziert und etliche illustrative Modellierungsbeispiele werden gegeben, die das breite Einsatzspektrum des vorgestellten Modellierungseinsatzes eindrucksvoll aufzeigen. Insbesondere die zahlreichen Typen von Task Units stellen wertvolle Hilfsmittel für die realitätsnahe Modellierung sehr unterschiedlicher Arten von Verarbeitungsvorgängen dar. Ohne einen Ansatz zur Modellierung der rechnerinternen Betriebsmittelverwaltung wäre es im Übrigen unmöglich,

hinreichend realistische Modelle von Software-Routern zu erstellen, die in der Lage sind, Verarbeitungsengpässe in dem Router zu identifizieren.

- Das sechste Kapitel beinhaltet eine Beschreibung der Integration des neuartigen Ansatzes von Herrn Runge zur Modellierung der rechnerinternen Betriebsmittelverwaltung in das Simulationswerkzeug ns-3. Insbesondere hat er ein dediziertes ns-3-Erweiterungsmodul realisiert und nutzt dabei Mehrfachvererbung. Auch sonstige wichtige Erweiterungen an ns-3 werden angesprochen, die Torsten Runge im Laufe seines Dissertations- bzw. des MEMPHIS-Projektes vorgenommen hat (z.B. im Hinblick auf eine realitätsnähere Generierung von Paketverkehr auf Basis gemessener Paket-Traces).
- In Kapitel 7 erarbeitet Herr Runge ein Wartenetzmodell für einen Software-Router, um den erzielbaren Maximaldurchsatz eines Routers zu prognostizieren. Insgesamt gesehen besitzt das Modell eine relativ geringe Komplexität und gestattet dennoch hinreichend realistische Prognosen bzgl. dem zu erwartenden Paketdurchsatz. Die Idee, die Paketbedienzeit im Router als eine lineare Funktion der Paketlänge anzunehmen, erweist sich dabei als erstaunlich erfolgversprechend und vereinfacht die Kalibrierung, da nur noch vier Parameter des Modells kalibriert werden müssen. Die Experimente zur Modellvalidierung, bei denen Durchsatzprognosen des Modells mit entsprechenden Durchsatzmessungen im realen Testbed verglichen werden, zeigen dann auch sehr gute Übereinstimmung zwischen dem Leistungsverhalten des Router-Modells sowie dem Router im realen Testbed. Die Untersuchung der Auswirkungen der Anzahl benutzter CPU-Kerne im Router u.a. auf Speed-up, Effizienz und parallelen Anteil ist ebenfalls recht aussagekräftig. Die Fallstudie zeigt, dass parallele Paketverarbeitung in Multi-Core-Systemen erfreulich gut skaliert.
- Die Fallstudie in Kapitel 8 untersucht, welchen Einfluss unterschiedliche Arten der Paketverarbeitung (wie Minimum Forwarding, IP-Routing und IPsec-Verschlüsselung) auf die Leistungsfähigkeit eines Software-Routers unter Linux haben. Für die verschiedenen Arten von Paketverarbeitung wird das unter Nutzung der ns-3-Erweiterung des Autors realisierte simulative Routermodell sorgfältig kalibriert und zumindest für *Minimum Forwarding*

auch erfolgreich validiert. Das Software-Router-Modell wurde dann in sehr umfangreichen Experimentserien für die drei betrachteten Paketverarbeitungsarten für die Prognose des zu erwartenden Paketdurchsatzes sowie der mittleren Paketlatenz im Router eingesetzt. In der Fallstudie dieses Kapitels wurde insbesondere die Paketankunftsrate im Router variiert bei zusätzlicher Veränderung der Paketlängen, die zum einen Poisson-Verkehr sowie zum anderen gemessenem Verkehr entsprechen (vgl. wohlbekannter Abilene-Trace).

- Das neunte Kapitel widmet sich nochmals einer wiederum sehr praxisrelevanten Fallstudie. Es wird versucht, Pakete in einem Software-Router in Abhängigkeit von den QoS-Anforderungen zu bearbeiten, welche die kommunizierenden Anwendungen besitzen, die für die Generierung der paketinhärenten Nutzdaten verantwortlich sind (z.B. Anwendungen mit Echtzeitkommunikationsanforderungen wie Audio-/Videoübertragungen oder aber Anwendungen wie Dateitransfer, die im Allgemeinen zeitunkritisch sind). Torsten Runge überlegt sich zunächst ein interessantes Konzept zur Priorisierung QoS-sensitiver Pakete mit Implikation der Priorisierung dieser Pakete gegenüber den nicht-zeitkritischen und daher einer bevorzugten Bearbeitung im Router infolge des Einsatzes unterschiedlicher Scheduling-Mechanismen bei der Paketverarbeitung. Das Router-Modell wird dann um das vorgeschlagene QoS-Konzept erweitert und dieses Modell wird wiederum mittels geeigneter Messdaten aus dem realen Testbed kalibriert und mittels mathematischer Berechnungen auch verifiziert (d.h. es wird nachgeprüft, ob das Modell das spezifizierte Verhalten aufweist). Die Übereinstimmung zwischen analytisch und simulativ erzielten Resultaten für die Latenzen der *Real-Time*- und der *Best-Effort*-Pakete sind extrem präzise. Zahlreiche Resultate zeigen schließlich in umfangreichen Simulationsstudien im Abschnitt 9.7 den großen Nutzen dieses Modells, u.a. um Schedulingstrategien wie prioritätsbasierte Paketbearbeitung im Detail mit den Strategien *Weighted Fair Queueing (WFQ)* und *Round Robin (RR)* in ihrer Auswirkung auf Interrupt-rate, Paketlatenzen, erzielbaren Maximaldurchsatz und CPU-Auslastung zu vergleichen.

Insgesamt gesehen ist der Innovationsgrad besonders eindrucksvoll, der im Bereich der Modellierung und Simulation dadurch erreicht wurde, dass ein neuartiges und erstaunlich allgemeingültiges Konzept zur Modellierung von Betriebsmittelverwaltung in kommunizierenden Rechnernetzknoten durch den Autor erarbeitet wurde und dies auch sehr erfolgreich in das Simulationswerkzeug ns-3 integriert werden konnte. Gekrönt wird diese Leistung dadurch, dass die ns-3-Erweiterung sogar als Open-Source-Software zur Verfügung steht.

Es ist zu hoffen, dass Arbeiten wie die vorliegende dazu beitragen werden, dass das innovative Gebiet der Software-Router zunehmend gut verstanden werden wird und dass das vielversprechende Potential dieser neuen Art von Netzkomponenten dadurch immer besser genutzt werden kann.

Hamburg, im Mai 2016

Bernd E. Wolfinger

Kurzfassung

Um Standard-Hardware in Kommunikationsnetzen bei hohem Verkehrsaufkommen einsetzen zu können und neben dem Senden und Empfangen auch erweiterte Datenverarbeitung (z.B. Verschlüsselung) vornehmen zu können, müssen Treiber, Netzwerkprotokollstapel und Anwendungen besser an die Hardware-Architektur und deren Parallelisierungsmöglichkeiten angepasst werden, um eine optimale Systemleistung zu erreichen. Trotz der vergleichsweise hohen Leistungsfähigkeit von Standard-Hardware stellt das Versenden, Empfangen und die Verarbeitung von Paketen bei hohem Verkehrsaufkommen eine große Herausforderung dar. Moderne Multi-Prozessor- bzw. Multi-Core-Systeme sind komplexe Systeme, deren Arbeitsabläufe und die dazu benötigten begrenzten Systemressourcen zu nicht trivial vorhersehbarem Leistungsverhalten führen. Die Identifizierung von Leistungsengpässen und die Untersuchung von Auswirkungen spezieller Verarbeitungsoptimierungen ist daher eine wichtige Voraussetzung zur Leistungssteigerung dieser Systeme und ist als solches ein bedeutendes Forschungsvorhaben.

In dieser Arbeit werden Methoden zur Leistungsbewertung und Optimierung der Paketverarbeitung auf aktuellen Multi-Core-Mikroprozessor-Systemen untersucht. Um die relevanten Leistungsengpässe in solchen komplexen Systemen zu verstehen und durch entsprechende Modifikationen optimieren zu können, wird ein allgemeiner Ansatz zur Modellierung der Datenverarbeitung und Ressourcenverwaltung in Rechnersystemen vorgeschlagen. Hierfür werden einfache aber hinreichend genaue analytische und simulative Modelle entworfen, welche das Zusammenspiel zwischen Hardware- (Prozessor, Netzwerkkarte, Hauptspeicher usw.) und Softwarekomponenten (Betriebssystem, Treiber usw.) beschreiben und die Ressourcenanforderungen der einzelnen Datenverarbeitungsschritte widerspiegeln. Der vorgeschlagene Modellierungsansatz ist generell für verschiedene Datenverarbeitungsszenarien anwendbar. Exemplarisch wurde dieses Konzept als Erweiterungsmodul **resource-management** für den in der Forschungsgemeinschaft weitverbreiteten Netzwerksimulator ns-3 implementiert. Dadurch wird das beschränkte ns-3-Knotenmodell erweitert und ermöglicht somit die detaillierte Modellierung der

rechnerinternen Datenverarbeitung sowie den Wettbewerb um Ressourcen. Auf Basis dieses ns-3-Erweiterungsmoduls werden spezielle Simulationsmodelle für verschiedene Arten von Netzwerknoten (z.B. Software Router) entworfen. Eine Kalibrierung und Validierung dieser Simulationsmodelle wird basierend auf realen Testbed-Messungen vorgenommen.

Mit Hilfe der validierten Simulationsmodelle werden verschiedene Fallstudien mit unterschiedlichem Verkehrsaufkommen (z.B. Bursts, Traces), Datenverarbeitungsfunktionalitäten (z.B. IP-Routing, IPsec-Verschlüsselung) und Rechnersystemkonfigurationen (z.B. Multi-Core, Treiberparameter) durchgeführt. Anhand von Leistungs- und Zuverlässigkeitskenngrößen (z.B. Maximaldurchsatz, Latenz, Paketverlust) werden die leistungsbeschränkenden Faktoren und Engpässe der untersuchten Systeme identifiziert. Des Weiteren ermöglichen es die Simulationsmodelle zu prognostizieren, wie sich verschiedene Modifikationen auswirken und Aussagen über Szenarien zu treffen, die mit den derzeit zur Verfügung stehenden Systemen noch nicht umsetzbar sind. Darauf basierend werden Optimierungen für Realsysteme zur Verbesserung der QoS-sensitiven Paketverarbeitung vorgeschlagen, welche außerdem als reine Softwarelösung kosteneffizient realisierbar sind. Insbesondere konnten dadurch Verbesserungen des aktuellen Stands der Technik bezüglich der Paketverarbeitung von QoS-sensitiven Netzapplikationen in Linux-basierten Software-Routern erzielt werden.

Abstract

In order to use standard hardware in high-speed communication networks and to be able to perform advanced data processing (e.g. encryption) – in addition to the data transmission and reception – the drivers, the network protocol stack and the applications need to be better adapted to the hardware architecture and their parallelization capabilities to achieve optimal system performance. Despite, the relatively high performance of standard hardware, the transmission, reception and processing of packets at heavy traffic loads represents a major challenge. Modern multi-processor or multi-core systems are complex systems, whose work processes and the required limited system resources lead to a performance behavior which prediction is non-trivial. Therefore, the identification of performance bottlenecks and the investigation of specific effects of optimizations is an important requirement to increase the performance of these systems and is as such an important research project.

In this thesis, methods are examined for current multi-core microprocessor systems for performance evaluation and optimization of packet processing. To understand the relevant performance bottlenecks in such complex systems and to be able to optimize them by appropriate modifications, a general approach to modeling the data processing and resource management in computer systems is proposed. Thus, simple but sufficiently accurate analytical and simulation models are designed, describing the interactions between hardware (processor, network card, main memory, etc.) and software components (operating system, drivers, etc.) and representing the resource requirements of specific data processing steps. The proposed modeling approach is generally applicable for various data processing scenarios. By way of example, this concept has been implemented as an extension module **resource-management** for the network simulator ns-3 which is widely used in the research community. This expands the ns-3 node model and enables detailed modeling of the system internal data processing as well as the resource contention. Based on this ns-3 extension module, specific simulation models for different types of network nodes are designed (e.g. software router). The calibra-

tion and validation of the simulation models is conducted based on real testbed measurements.

With the help of validated simulation models several case studies with different traffic loads (e.g. bursts, traces), data processing functionalities (e.g. IP routing, IPsec encryption) and computer system configurations (e.g. multi-core, driver parameters) are carried out. On the basis of performance and reliability characteristics (e.g. maximum throughput, latency, packet loss), the performance-limiting factors and bottlenecks of the analyzed systems are identified. Furthermore, the simulation models enable predictions how various modifications influence the system performance and they give the opportunity to make evaluations about scenarios that are not feasible with the currently available systems. Based on that, optimizations are proposed for real systems to improve the QoS-sensitive packet processing, which are also implementable in a cost-efficient manner as a pure software solution. In particular, thereby an improvement of the current state of the art could be achieved with respect to the packet processing of latency-sensitive network applications in Linux-based software routers.

Danksagung

Eine wissenschaftliche Arbeit ist selten das Werk einer einzelnen Person, deshalb möchte ich mich bei allen Menschen bedanken, ohne welche diese Arbeit nicht zu diesem Werk geworden wäre.

An erster Stelle bin ich meinem Doktorvater Prof. Bernd E. Wolfinger zu besonderen Dank verpflichtet, welcher mich stets durch seine langjährigen Erfahrungen mit zahlreichen Anmerkungen und Ratschlägen gefördert und gefordert hat. Dadurch konnte ich viel Wissen insbesondere in den Bereichen der analytischen Modellierung und Simulation zur Leistungsbewertung von Rechnersystemen und Rechnernetzen aufbauen.

Ebenso danke ich meinen Kollegen Alexander Beifuß, Dr. Stephan Heckmüller, Dr. Klaus-Dieter Heidtmann, Andrey Kolesnikov, Dr. Martin Lehmann und Robert Olotu in der Arbeitsgruppe Telekommunikation und Rechnernetze der Universität Hamburg, welche mich mit vielen bereichernden Hinweisen und Diskussionen immer wieder auf neue Themen gelenkt haben. Für die erfolgreiche Zusammenarbeit im Rahmen von studentischen Arbeiten danke ich außerdem Alexander Beifuß, Malte Hamann, Paul Lindt und Norbert Schramm.

Des Weiteren möchte ich mich bei den Kollegen Prof. Georg Carle, Florian Wohlfart, Daniel Raumer, Paul Emmerich und Dr. Lothar Braun vom Lehrstuhl für Netzarchitekturen und Netzdienste der TU München für die fruchtbare Zusammenarbeit innerhalb des DFG-Projekts MEMPHIS bedanken, welche meine Arbeit durch intensive Diskussionen sowie die Bereitstellung von vielzähligen Testbed-messungen sehr unterstützt haben.

Den Gutachtern Prof. Thomas Ludwig und Prof. Otto Spaniol möchte ich meinen Dank für ihre wissenschaftliche Einschätzung der Arbeit aussprechen.

Meinen Eltern Regina und Lothar Meyer danke ich für ihre fortwährende Unterstützung, welche mir stets in allen Lebenslagen zur Seite stehen.

Schließlich möchte ich meiner Frau Dr. Ulrike Runge von ganzem Herzen für ihre unermüdliche Unterstützung und liebevollen Ermutigungen danken.

Inhaltsverzeichnis

I	Einführung und Grundlagen	1
1	Einführung	3
1.1	Motivation	3
1.2	Projekt MEMPHIS	5
1.3	Beiträge der Arbeit	6
1.4	Gliederung der Arbeit	6
2	Grundlagen und Verwandte Arbeiten	9
2.1	Definitionen	10
2.2	Mooreches Gesetz	11
2.3	Amdahlsches Gesetz	12
2.4	Parallelitätsebenen	14
2.5	Klassifizierung von Rechnerarchitekturen	17
2.6	Verwandte Arbeiten	19
2.6.1	Paketverarbeitung auf Standard-Hardware	19
2.6.2	Modellierung von Rechnersystemen	22

II	Datenverarbeitung in Rechnersystemen	29
3	Hardware zur Datenverarbeitung	31
3.1	Mikroprozessor-Architektur	32
3.1.1	Hauptspeicher und Cache	36
3.1.2	Bus-Systeme	41
3.2	Netzwerkkarte	42
3.2.1	Receive-Side Scaling und Multi-Queueing	42
3.2.2	Netzwerkkarten-basierte Paketklassifizierung	44
3.2.3	Interrupts	45
3.3	Sonstige Hardware-Architekturen	46
4	Software zur Paketverarbeitung	49
4.1	Überblick zur Paketverarbeitung	50
4.2	Betriebssysteme zur Paketverarbeitung	52
4.2.1	Netzwerk-Betriebssysteme	55
4.2.2	Echtzeit-Betriebssysteme	56
4.3	Paketverarbeitung im Linux-Betriebssystem	57
4.3.1	Interrupt-basierte Paketverarbeitung	57
4.3.2	NAPI-basierte Paketverarbeitung	58
4.3.3	Treiber-basierte Paketverarbeitung	63
4.4	Softwareoptimierungen für die Paketverarbeitung	65
4.4.1	Parallelisierung der Paketverarbeitung	65
4.4.2	Batching	68
4.4.3	Präallokation und Recycling	69
4.4.4	Affinität	70
4.4.5	Prefetching	70
4.4.6	Memory Mapping und Zero-Copy	71
4.5	Networking Frameworks zur Paketverarbeitung	72
4.5.1	Click Modular Router	73
4.5.2	DPDK	75
4.5.3	Netmap	76
4.5.4	PF_RING DNA	77

III	Modellierung von Rechnersystemen	79
5	Konzeption eines Modellierungsansatzes für Rechnersysteme	81
5.1	Motivation zur Modellierung von Rechnersystemen	
	in Kommunikationsnetzen	82
5.2	Modellierungsanforderungen	86
5.2.1	Funktionale Anforderungen	86
5.2.2	Nichtfunktionale Anforderungen	88
5.3	Überblick zum Modellierungsansatz	89
5.4	Task Unit	91
5.4.1	Verhalten der Task Unit	93
5.4.2	Strategien der Task Unit	96
5.4.3	Task Unit Typen	97
5.4.4	Task Unit Graphen	101
5.5	Resource Manager	103
5.5.1	Verhalten des Resource Managers	104
5.5.2	Scheduling-Strategien des Resource Managers	106
5.5.3	Resource Pool und Ressourcen	108
5.6	Interaktionen Resource Manager und Task Unit	109
5.7	Fallbeispiel: Paketverarbeitung	113
5.8	Zusammenfassung	115
6	Implementierung des Modellierungsansatzes für Rechnersysteme	117
6.1	Integration des Resource Managements	118
6.2	Implementierungskonzept	120
6.3	Implementierung der Task Unit	121
6.4	Implementierung des Resource Managers	123
6.5	Implementierung der Resource	125
6.6	Sonstige ns-3-Erweiterungen	126
6.6.1	Poisson-Verkehr	126
6.6.2	Unterstützung von Type of Service	127
6.6.3	Trace-basierter Verkehr	127
6.7	Zusammenfassung	128

IV Fallstudien zur Paketverarbeitung in Rechnersystemen 131

7 Fallstudie: Skalierung der Paketverarbeitung	133
7.1 Modellentwurf	134
7.2 Modellkalibrierung	138
7.3 Modellvalidation	141
7.4 Auswertung der Resultate	143
7.4.1 Einfluss der Frame-Größe	144
7.4.2 Einfluss der Anzahl der CPU-Kerne	146
7.5 Zusammenfassung	150
8 Fallstudie: Paketverarbeitungsarten	151
8.1 Modellentwurf	152
8.2 Modellkalibrierung	156
8.2.1 Minimal Forwarding	158
8.2.2 IP-Routing	159
8.2.3 IPsec-Verschlüsselung	160
8.3 Modellvalidation	161
8.4 Auswertung der Resultate	163
8.4.1 Durchsatz	165
8.4.2 Paketlatenz	167
8.4.3 Paketverlust	169
8.5 Zusammenfassung	172
9 Fallstudie: QoS-sensitive Paketverarbeitung	173
9.1 QoS-Unterstützung im Linux-Betriebssystem	174
9.2 Konzept zur QoS-sensitiven Paketverarbeitung	176
9.2.1 Verkehrsklassifizierung	177
9.2.2 Verkehrspriorisierung	178
9.3 Modellentwurf	180
9.3.1 Modellierung der Linux NAPI mit Task Units	182
9.3.2 Modellannahmen und Limitierungen	185
9.4 Modellkalibrierung	186
9.5 Modellverifizierung	191

9.6	Auswertung der Resultate	195
9.6.1	Einfluss des Echtzeit-Anteils auf den Maximaldurchsatz . .	197
9.6.2	Einfluss des Echtzeit-Anteils auf die Paketlatenz	198
9.6.3	Einfluss der Last bei konstantem Echtzeit-Anteil	201
9.7	Zusammenfassung	205
V	Fazit und Ausblick	207
10	Fazit und Ausblick	209
10.1	Fazit	209
10.2	Ausblick	213
	Abkürzungsverzeichnis	217
	Abbildungsverzeichnis	221
	Tabellenverzeichnis	225
	Literaturverzeichnis	227

“Wer hohe Türme bauen will, muss lange beim Fundament verweilen.“

— Anton Bruckner (1824-1896),
österreichischer Komponist