

Diss. ETH No. 22149
TIK-Schriftenreihe Nr. 149

Dynamic Protocol Stacks

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Science

presented by

ARIANE TRAMMELL-KELLER

Master of Science ETH in Electrical Engineering and
Information Technology

born April 10, 1982

citizen of Sarmenstorf (AG)

accepted on the recommendation of
Prof. Dr. Bernhard Plattner, examiner
Prof. Dr. Marco Platzner, co-examiner
Dr. Martin May, co-examiner

2014

Berichte aus der Kommunikationstechnik

Ariane Trammell

Dynamic Protocol Stacks

Shaker Verlag
Aachen 2014

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Zugl.: Zürich, ETH, Diss., 2014

Copyright Shaker Verlag 2014

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8440-3114-0

ISSN 0945-0823

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen

Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9

Internet: www.shaker.de • e-mail: info@shaker.de

Abstract

Many people around the globe enjoy the amenities of a ubiquitously available communication infrastructure. This infrastructure is used in a variety of scenarios: It is used for personal communication with friends or business partners, to obtain information or entertainment from a service provider, or for buying things without being physically present in a shop. Additionally, the same infrastructure is used by electronic devices to share information. For example, sensor nodes measure temperature, humidity, air pollution etc., and share this information with a central server which gathers aggregated knowledge from the individual measurement points. Or different local sites of a company might exchange a high data volume to synchronize and / or back up their data base.

Each of these applications impose different communication requirements on the infrastructure, ranging from different levels of security and privacy, to different allowable transmission delays, to different requirements on the transmission reliability. However, all these applications run over the same network architecture, the Internet architecture. Currently, many add-ons to the original architecture make this possible. However, it is questionable whether this add-on approach is scalable and can accommodate future applications. Therefore, many researchers develop new network architectures from scratch, so called clean slate architectures, that are designed to be more appropriate for future challenges.

In this dissertation we have developed such a clean slate network architecture called *Dynamic Protocol Stack (DPS) architecture*. The DPS architecture tackles the challenge of providing a future-oriented architecture by introducing flexibility at two levels. First, newly

developed communication protocols can be included seamlessly in the architecture and second, the protocol stack used for the communication can be adapted to the current environment and communication needs while transferring data.

However, such a flexible network architecture poses a new challenge: How can it be implemented without suffering from performance degradation introduced by the flexibility? We address this challenge by providing two different implementations of the DPS architecture. With the first implementation we show how the DPS architecture can be implemented on a general-purpose CPU, and we show that it has the same performance as an implementation of the Internet architecture on the same system. With the second implementation we show how the DPS architecture can make use of hardware acceleration for the individual network protocols. Therefore, we implemented the DPS architecture on an FPGA-based system-on-chip platform. This implementation, called EmbedNet, allows for the dynamic run-time mapping of network functionality to either software or hardware. Thus it offers hardware acceleration for arbitrary network functionality in the DPS architecture as dedicated hardware for a specific network protocol would in the Internet architecture.

Additionally, we show in two use cases the benefits of having a dynamic network architecture. First, we show how we can minimize protocol overhead by adapting the protocol stack to the current network conditions. Second, we show how we can improve the system performance when adapting the hardware / software mapping to the current network traffic. Finally, we discuss how more advanced optimization algorithms could be developed to optimize the provided network functionality.

To summarize, with this dissertation we have shown that it is feasible to provide dynamic network architectures and our results suggest that it is worthwhile to go further in that direction.

Kurzfassung

Viele Menschen auf der ganzen Welt schätzen es, sich jederzeit auf eine zuverlässige Kommunikationsinfrastruktur verlassen zu können. Diese Infrastruktur wird in den vielfältigsten Szenarien verwendet: Für die Kommunikation mit Freunden oder Geschäftspartnern, um sich zu informieren, um einzukaufen, ohne in einen Laden zu gehen oder einfach aus Spass. Diese Infrastruktur wird auch von elektronischen Geräten gebraucht um gegenseitig Informationen auszutauschen. Zum Beispiel können Sensoren Umweltdaten wie Temperatur, Luftfeuchtigkeit oder Luftverschmutzung messen und an einen zentralen Rechner schicken, der diese Messpunkte dann auswertet. Ein anderes Beispiel ist das Verschicken von grossen Datenmengen zwischen verschiedenen Firmenstandpunkten, um ihre Datenbanken abzugleichen oder um die Daten zu sichern.

All diese Anwendungen stellen die verschiedensten Anforderungen an die Kommunikationsinfrastruktur, wie zum Beispiel unterschiedliche Bedürfnisse zum Sichern der übertragenen Informationen, verschiedene maximal zulässige Übertragungszeiten oder verschiedene Anforderungen an die Zuverlässigkeit der Datenübertragung.

Trotz der unterschiedlichen Anforderungen verwenden alle Anwendungen dieselbe Netzwerkarchitektur: die Internet-Architektur. Im Laufe der letzten Jahre wurde die Internet-Architektur an vielen Stellen erweitert, um neuartigen Anforderungen gerecht werden zu können. Es ist aber fraglich, ob dieser Ansatz auch in Zukunft den Anforderungen immer neuer Anwendungen genügen kann. Deshalb beschäftigen sich viele Forscher damit, radikal neue Kommunikationsarchitekturen zu entwerfen. Diese Architekturen sollen besser auf zukünftige Anforderungen reagieren können.

In dieser Doktorarbeit haben wir eine derartige Netzwerkarchitektur entwickelt. Sie trägt den Namen *Dynamic Protocol Stack (DPS)* Architektur. Die DPS-Architektur verfolgt folgende zwei Ansätze um den zukünftigen Anforderungen zu genügen: Erstens können neu entwickelte Protokolle nahtlos in die Architektur eingefügt werden und zweitens kann die Auswahl der zur Datenübertragung verwendeten Protokolle während einer Kommunikation angepasst werden, um den Kommunikationsanforderungen in jeder Situation gerecht zu werden.

Solch eine flexible Netzwerk-Architektur stellt allerdings neue Herausforderungen: Wie kann sie umgesetzt werden, ohne dass die Leistungsfähigkeit durch die Flexibilität beeinträchtigt wird? Wir zeigen dies anhand zweier Umsetzungen. Bei der ersten Umsetzung implementieren wir die DPS-Architektur auf einem Prozessor. Diese Umsetzung kann die DPS-Pakete gleich schnell verarbeiten wie Pakete der Internet-Architektur. Bei der zweiten Umsetzung machen wir uns Hardware-Beschleuniger zu Nutze, um die Funktionen der einzelnen Netzwerkprotokolle zu beschleunigen. Dazu haben wir die DPS-Architektur auf einem FPGA-basierten System implementiert. Diese Umsetzung, die wir EmbedNet nennen, ermöglicht es, einzelne Netzwerkfunktionen entweder in Hardware oder in Software auszuführen. Die dynamische Zuordnung zu Hardware oder Software kann während der Kommunikation in Echtzeit verändert werden. Auf diese Weise kann sich die Laufzeitumgebung von EmbedNet für verschiedene Kommunikationsszenarien während der Datenübertragung optimieren.

Mit der Untersuchung zweier Anwendungsfälle zeigen wir die Vorzüge einer flexiblen Netzwerkarchitektur auf. Zuerst präsentieren wir, wie man die Kommunikationskosten minimieren kann, indem man die verwendeten Protokolle sich ändernden Netzwerkbedingungen kontinuierlich anpasst. Zudem präsentieren wir, wie der Systemdurchsatz verbessert werden kann, wenn man die Zuordnung von Netzwerkfunktionen zu Hardware oder Software dem gegenwärtigen Netzwerkverkehr anpasst. Schliesslich zeigen wir auf, wie man die Optimierungsalgorithmen verbessern könnte, um noch bessere Ergebnisse zu erzielen.

Zusammenfassend verdeutlichen wir mit dieser Doktorarbeit, dass es möglich ist, dynamische Netzwerkarchitekturen zu entwickeln und dass es vielversprechend ist, weiter in diese Richtung zu forschen.

Contents

Abstract	iii
Kurzfassung	v
1 Introduction	1
1.1 From Active Networking to Future Internet Research . .	2
1.2 Flexible Network Architectures	4
1.3 Contributions	7
1.4 Scope	8
1.5 Thesis Structure	9
 I Flexible Network Architectures	 11
2 The Autonomic Network Architecture	13
2.1 Node Architecture	13
2.2 Protocol Stack Setup	15
2.3 Node Local Protocol Stack Adaptation	18
2.4 Inter Node Protocol Stack Adaptation	20
 3 The Dynamic Protocol Stack Architecture	 23
3.1 Node Architecture	23
3.2 Protocol Setup	25
3.3 Node Local Protocol Stack Adaptation	33
3.4 Inter Node Protocol Stack Adaptation	34

4	Comparison of ANA and the DPS Architecture	37
5	Section Conclusion	41
II	Execution Environments for Dynamic Protocol Stacks	43
6	Packet Processing Execution Environments	45
7	DPS on a General-Purpose Processor	49
7.1	Architecture	49
7.2	Performance Evaluation	56
8	DPS with Hardware Accelerators	59
8.1	EmbedNet Architecture	60
8.2	EmbedNet Hardware Design	66
8.3	Performance Evaluation	73
9	Section Conclusion	81
III	Case Studies with Dynamic Adaptaion	83
10	Autonomic Adaptation of the Protocol Stack Functionality	85
11	Autonomic HW/SW Mapping in EmbedNet	91
11.1	Mapping Algorithms	93
11.2	Network Traffic Mixes	94
11.3	Experimental Results	94
12	Section Conclusion	101
IV	Towards Self-Awareness in Networking	103
13	Adaptation Strategies	105

14 Self-Awareness in Networking	109
14.1 Proprioceptive Framework	109
14.2 Levels of Self-Awareness	111
15 Section Conclusion	115
 V Conclusions	 117
16 Comparison with Related Work	119
16.1 Future Internet Architectures	119
16.2 Network Processing with FPGAs	122
17 Review of Contributions	129
18 Critical Assessment	135
19 Future Work and Outlook	139
Abbreviations	163
Acknowledgements	167
Curriculum Vitae	171

