Informatik aktuell

Herausgeber: W. Brauer
im Auftrag der Gesellschaft für Informatik (GI)

Torsten Braun
Georg Carle
Burkhard Stiller (Hrsg.)

Kommunikation in Verteilten Systemen (KiVS)

15. Fachtagung Kommunikation in Verteilten Systemen (KiVS 2007)
Bern, Schweiz, 26. Februar – 2. März 2007

Eine Veranstaltung der
Informationstechnischen Gesellschaft (ITG/VDE)
unter Beteiligung der Gesellschaft für Informatik (GI)
Ausgerichtet von der Universität Bern, Schweiz

ITG INFORMATIONSTECHNISCHE GESELLSCHAFT IM VDE

Springer

Herausgeber

Torsten Braun
Universität Bern, Institut für Informatik u. Angewandte Mathematik
Neubrückstr. 10, CH-3012 Bern

Georg Carle
Universität Tübingen, Wilhelm-Schickard-Institut für Informatik
Sand 13, D-72076 Tübingen

Burkhard Stiller
Universität Zürich, Institut für Informatik
Binzmühlestr. 14, CH-8050 Zürich

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

CR Subject Classification (1998):
B.4, C.2, K.6.4, K.6.5, D.4.4, D.4.6, C.4, H.4.3, K.6.5

ISSN 1431-472-X
ISBN-10 3-540-69961-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-69961-3 Springer Berlin Heidelberg New York

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Springer Berlin Heidelberg New York
Springer ist ein Unternehmen von Springer Science+Business Media

springer.de

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Satz: Reproduktionsfertige Vorlage vom Autor/Herausgeber
Gedruckt auf säurefreiem Papier SPIN: 11980728 33/3142-543210

Vorwort

Die 15. ITG/GI-Fachtagung „Kommunikation in Verteilten Systemen" (KiVS 2007) wird vom 26. Februar bis 2. März 2007 an der Universität Bern (Schweiz) durchgeführt. Die in einem zweijährigem Rhythmus stattfindende KiVS wird damit zum ersten Mal in ihrer 29-jährigen Geschichte außerhalb Deutschlands veranstaltet.

Die KiVS zeichnet sich durch eine große Breite im Bereich der Kommunikation und verteilten Systeme aus. Mit den Overlay- und Peer-to-Peer-Netzen hat sich ein wichtiger Schwerpunkt entwickelt, der bereits auf der letzten KiVS 2005 in Kaiserslautern mit einem Workshop vertreten war. Im Rahmen der KiVS 2007 wird nun ein spezielles Peer-to-Peer-Symposium durchgeführt. Das Konzept der Selbstorganisation ist nicht nur eng mit Peer-to-Peer-Netzen verbunden, sondern bildet auch für Sensornetzen und mobile Ad-Hoc-Netze eine wichtige Grundlage. Verteilte Anwendungen wie Web Services sind auf gut funktionierende Kommunikationsnetze, welche die gewünschte Dienstgüte und Robustheit erbringen müssen, sowie auf sichere Kommunikationssysteme angewiesen.

Die genannten Themen werden während der KiVS 2007 durch verschiedenartige Beiträge aus der Forschung diskutiert. Aus 74 eingereichten Forschungsbeiträgen hat das Programmkomitee auf seiner Sitzung in Tübingen zwanzig Artikel ausgewählt, welche in diesem Tagungsband präsentiert werden. Außerdem werden traditionell an der KiVS die Preise für die besten Dissertationen und Diplom- bzw. Master-Arbeiten der vergangenen Jahre vergeben. Neun Beiträge der Preisträger aus den Jahren 2005 und 2006 runden diesen Tagungsband ab.

Das Tagungsprogramm wird durch Exkursionen (CERN Genf, IBM Zürich, Swisscom Bern), Kurz- und Industriebeiträge, Tutorien, eingeladenen Vorträgen und Workshops ergänzt. Kurzbeiträge diskutieren neuartige Forschungsideen, Industriebeiträge beschreiben aktuelle Projekte, Technologien sowie heutige Trends aus der industriellen Praxis. Die Workshops vertiefen verschiedene Themengebiete: Workshop zu Mobilen Ad-Hoc Netzwerken; Netzwerksicherheit: Sichere Netzwerkkonfiguration; Selbstorganisierende, adaptive, kontextsensitive verteilte Systeme; Service-oriented Architectures und Service-oriented Computing.

Neben eingeladenen Sprechern aus Industrie und Forschung konnten vier Tutorien zu hochaktuellen Themen in Forschung und Praxis gewonnen werden: Self-Organization in Autonomous Sensor/Actuator Networks, Network Calculus – A Life After IntServ, Economics-Informed Network Design und Web 2.0.

Die KiVS ist ohne die tatkräftige Unterstützung vieler Personen und Organisationen in dieser Form nicht denkbar. Wir danken allen Sponsoren sowie Organisation und Verbänden für Ihre wertvolle Unterstützung. Den Mitarbeiterin-

nen und Mitarbeitern unserer Forschungsgruppen an den Universitäten Bern, Tübingen und Zürich danken wir für ihren enormen Einsatz zur Vorbereitung, Organisation und Durchführung der Tagung.

Bern, im Februar 2007

Torsten Braun
Georg Carle
Burkhard Stiller

Organisation

Die 15. ITG/GI Fachtagung „Kommunikation und Verteilte Systeme“ (KiVS 2007) wurde von den Professoren Braun (Universität Bern), Carle (Universität Tübingen) und Stiller (Universität Zürich) sowie deren Forschungsgruppen organisiert und in Kooperation mit der Informationstechnischen Gesellschaft des VDE unter Beteiligung der Gesellschaft für Informatik an der Universität Bern veranstaltet.

Tagungsleitung

Torsten Braun (Universität Bern)
Georg Carle (Universität Tübingen)
Burkhard Stiller (Universität Zürich)

Programmkomitee

S. Abeck	Universität Karlsruhe
K. Aberer	Ecole Polytechnique Fédérale de Lausanne
H. Baldus	Philips Research Aachen
F. Baumgartner	Swisscom
C. Becker	Universität Mannheim
T. Braun	Universität Bern
R. Breu	Universität Innsbruck
B. Butscher	Fraunhofer FOKUS Berlin
G. Carle	Universität Tübingen
J. Charzinski	Siemens AG
K. David	Universität Kassel
H. de Meer	Universität Passau
J. Eberspächer	Technische Universität München
W. Effelsberg	Universität Mannheim
A. Feldmann	Deutsche Telekom Laboratories / TU Berlin
A. Ferscha	Universität Linz
S. Fischer	Universität zu Lübeck
K. Geihs	Universität Kassel
C. Görg	Universität Bremen
R. Gotzhein	Universität Kaiserslautern
C. Graf	Switch
G. Hasslinger	Deutsche Telekom
B. Haverkort	Universität Twente
O. Heckmann	Technische Universität Darmstadt
H. G. Hegering	Universität München
R. G. Herrtwich	Daimler-Chrysler

K. Irmscher	Universität Leipzig
M. Kaiserswerth	IBM
H. Karl	Universität Paderborn
P. Kaufmann	DFN-Verein
R. Keller	Ericsson Research
W. Kellerer	DoCoMo EuroLaboratories
U. Killat	TU Hamburg-Harburg
H. König	Brandenburgische Technische Universität Cottbus
U. Krieger	Universität Bamberg
P. Kropf	Université de Neuchâtel
P. J. Kühn	Universität Stuttgart
W. Lamersdorf	Universität Hamburg
R. Lehnert	Technische Universität Dresden
S. Leinen	Switch
H. Leopold	Telekom Austria
C. Lindemann	Universität Leipzig
C. Linnhoff-Popien	Ludwig-Maximilians-Universität München
N. Luttenberger	Universität Kiel
F. Mattern	Eidgenössische Technische Hochschule Zürich
P. Müller	Universität Kaiserslautern
B. Plattner	Eidgenössische Technische Hochschule Zürich
C. Prehofer	Nokia Research Center
E. P. Rathgeb	Universität Duisburg-Essen
P. Reichl	Forschungszentrum Telekommunikation Wien
H. Ritter	Freie Universität Berlin
K. Rothermel	Universität Stuttgart
G. Schäfer	Technische Universität Ilmenau
A. Schill	Technische Universität Dresden
J. Schiller	Freie Universität Berlin
J. Schmitt	Universität Kaiserslautern
O. Spaniol	Rheinisch-Westfälische Technische Hochschule Aachen
R. Steinmetz	Technische Universität Darmstadt
B. Stiller	Universität Zürich
H. Stüttgen	NEC Europa
P. Tran-Gia	Universität Würzburg
C. Tschudin	Universität Basel
U. Ultes-Nitsche	Université de Fribourg
K. Wehrle	Rheinisch-Westfälische Technische Hochschule Aachen
L. C. Wolf	Technische Universität Braunschweig
B. Wolfinger	Universität Hamburg
A. Wolisz	Technische Universität Berlin
M. Zitterbart	Universität Karlsruhe

Sponsoren und unterstützende Organisationen

Business Network Network Communications
Cisco Systems
Ericsson
IBM
NEC
Nokia
Philips
Swisscom
SWITCH
Telekom Austria
VISCOM Visual Communications
Whitestein Technologies

German Chapter of the ACM
Gesellschaft für Informatik, Fachgruppe Kommunikation und Verteilte Systeme
Gesellschaft für Informations- und Kommunikationstechnik des Österreichischen Verbands für Elektrotechnik
IEEE Schweiz
Informationstechnische Gesellschaft im VDE
Informationstechnische Gesellschaft von Electrosuisse
Schweizer Informatik Gesellschaft
Schweizerischer Technischer Verband, Fachgruppe Elektronik und Informatik
Telematik-Cluster Bern

Inhaltsverzeichnis

I Overlay und Peer-to-Peer Networks

Netzwerkeffizienz stabiler Overlay-Streaming-Topologien 3
Thorsten Strufe, Jens Wildhagen, Günter Schäfer (Technische Universität Ilmenau)

Improving the Performance and Robustness of Kademlia-Based Overlay Networks 15
Andreas Binzenhöfer, Holger Schnabel (Universität Würzburg)

Improved Locality-Aware Grouping in Overlay Networks 27
Matthias Scheidegger, Torsten Braun (Universität Bern)

On Improving the Performance of Reliable Server Pooling Systems for Distance-Sensitive Distributed Applications 39
Thomas Dreibholz, Erwin P. Rathgeb (Universität Duisburg-Essen)

Modeling Trust for Users and Agents in Ubiquitous Computing 51
Sebastian Ries, Jussi Kangasharju, Max Mühlhäuser (Technische Universität Darmstadt)

A Decentral Architecture for SIP-based Multimedia Networks 63
Holger Schmidt, Teodora Guenkova-Luy, Franz J. Hauck (Universität Ulm)

II Verteilte Anwendungen und Web Services

An Orchestrated Execution Environment for Hybrid Services 77
Sandford Bessler, Joachim Zeiss, Rene Gabner (Forschungszentrum Telekommunikation Wien), Julia Gross (Kapsch CarrierCom)

A Lightweight Service Grid Based on Web Services and Peer-to-Peer 89
Markus Hillenbrand, Joachim Götze, Ge Zhang, Paul Müller (Universität Kaiserslautern)

Semantic Integration of Identity Data Repositories 101
Christian Emig, Kim Langer, Sebastian Abeck (Universität Karlsruhe), Jürgen Biermann (iC Consult)

Throughput Performance of the ActiveMQ JMS Server 113
Robert Henjes, Daniel Schlosser, Michael Menth, Valentin Himmler (Universität Würzburg)

III Sensornetze und Mobile Ad-Hoc-Netze

Titan: A Tiny Task Network for Dynamically Reconfigurable Heterogeneous Sensor Networks 127
Clemens Lombriser, Daniel Roggen, Mathias Stäger, Gerhard Tröster (Eidgenössische Technische Hochschule Zürich)

Key Exchange for Service Discovery in Secure Content Addressable Sensor Networks 139
Hans-Joachim Hof, Ingmar Baumgart, Martina Zitterbart (Universität Karlsruhe)

NIDES: Ein Verfahren zur Multihop-Distanzschätzung mittels Nachbarschaftsanalyse 151
Carsten Buschmann, Christian Werner, Horst Hellbrück, Stefan Fischer (Universität Lübeck)

Verhaltensbeobachtung und -bewertung zur Vertrauensbildung in offenen Ad-hoc-Netzen 163
Daniel Kraft (Universität Karlsruhe), Günter Schäfer (Technische Universität Ilmenau)

IV Dienstgüte und Sicherheit

A Priori Detection of Link Overload due to Network Failures 177
Jens Milbrandt, Michael Menth, Frank Lehrieder (Universität Würzburg)

Analysis of a Real-Time Network Using Statistical Network Calculus with Approximate Invariance of Effective Bandwidth 189
Kishore Angrishi, Shu Zhang, Ulrich Killat (Technische Universität Hamburg-Harburg)

Comparison of Preemptive and Preserving Admission Control for the UMTS Enhanced Uplink 201
Andreas Mäder, Dirk Staehle (Universität Würzburg)

A New Model for Public-Key Authentication 213
Reto Kohlas, Jacek Jonczy, Rolf Haenni (Universität Bern)

A Proof of Concept Implementation of SSL/TLS Session-Aware User Authentication (TLS-SA) 225
Rolf Oppliger (eSECURITY Technologies), Ralf Hauser (PrivaSphere), David Basin (Eidgenössische Technische Hochschule Zürich), Aldo Rodenhaeuser, Bruno Kaiser (AdNovum)

Secure TLS: Preventing DoS Attacks with Lower Layer Authentication .. 237
Lars Völker (Universität Karlsruhe),
Marcus Schöller (University of Lancaster)

V Preisträger

Traffic Shaping in a Traffic Engineering Context 251
Dirk Abendroth (BMW AG)

Routing and Broadcasting in Ad-Hoc Networks 259
Marc Heissenbüttel (Universität Bern)

Benutzerorientierte Leistungs- und Verfügbarkeitsbewertung von Internetdiensten am Beispiel des Portals hamburg.de 267
Martin Gaitzsch (Universität Hamburg)

A System for in-Network Anomaly Detection 275
Thomas Gamer (Universität Karlsruhe)

Simulation-Based Evaluation of Routing Protocols for Vehicular Ad Hoc Networks 283
Sven Lahde (Technische Universität Braunschweig)

Dynamic Algorithms in Multi-user OFDM Wireless Cells 291
James Gross (Technische Universität Berlin)

Self-Organizing Infrastructures for Ambient Services 299
Klaus Herrmann (Universität Stuttgart)

Bereitstellung von Dienstgüte für aggregierte Multimedia-Ströme in lokalen 'Broadcast'-Netzen 307
Stephan Heckmüller (Universität Hamburg)

Ein heuristisches Verfahren zur dienstgütebasierten Optimierung flexibler Geschäftsprozesse 315
Michael Spahn (Technische Universität Darmstadt)

Index der Autoren 323

Teil I

Overlay und Peer-to-Peer Networks

Netzwerkeffizienz stabiler Overlay-Streaming-Topologien

Thorsten Strufe[1], Jens Wildhagen[2] und Günter Schäfer[1]

[1] Fachgebiet Telematik/Rechnernetze, Technische Universität Ilmenau
[2] Fachgebiet Integrierte Hard- und Softwaresysteme, Technische Universität Ilmenau

Zusammenfassung. Bei der Konstruktion von Overlay-Topologien für multimediale Live-Streaming-Anwendungen sind zwei Eigenschaften von besonderer Bedeutung: die *Netzwerkeffizienz* der Topologie in Bezug auf die Paketverteilung und die *Stabilität* der Topologie sowohl im Fall vorsätzlicher Sabotageangriffe als auch bei zufälligen Knotenausfällen. Während ein Großteil der existierenden Ansätze hauptsächlich Effizienzkriterien optimiert und nur wenige Ansätze Stabilität gegen zufällige Ausfälle betrachten, ist es uns in früheren Arbeiten gelungen, Verfahren für die Konstruktion angriffsstabiler Topologien zu entwickeln [17,3]. Da in diesen Arbeiten zum Zweck der Steigerung der Stabilitätseigenschaften der Topologie eine Verschlechterung der Effizienzeigenschaften bewusst in Kauf genommen wurde, wird in dem vorliegenden Artikel ein Verfahren vorgestellt, dass es ermöglicht, einen Kompromiss zwischen Effizienz und Stabilität bei der Konstruktion der Topologie zu finden.
Hierzu werden zunächst Stabilitäts- und Effizienzeigenschaften in Form von Kostenmetriken operationalisiert und darauf aufbauend ein verteilter Algorithmus zur dynamischen Topologieoptimierung vorgestellt, der eine Gesamtkostenfunktion optimiert, die durch eine parametrisierbare, gewichtete Kombination der Einzelmetriken definiert ist. Mit Hilfe einer Simulationsstudie wird gezeigt, dass auf diese Weise gute Kompromisse zwischen Effizienz und Stabilität bei der Topologiekonstruktion gefunden werden können.

1 Einleitung

In den vergangenen Jahren wurde für den Internet-basierten Transport von Multimedia-Inhalten hin zu großen Benutzerpopulationen der Einsatz des sogenannten *"Application Level Multicast"*-Ansatzes (ALM) vorgeschlagen [7]. Der prinzipielle Vorteil dieses Ansatzes ist, dass die den Datenstrom empfangenden Systeme diesen für andere Systeme replizieren und somit das "Angebot" an potentiellen Quellen für einen bestimmten Datenstrom automatisch mit der Nachfrage nach diesem Datenstrom steigt. Der Ansatz weist daher theoretisch eine beliebige Skalierbarkeit in der Anzahl der Empfänger auf.

Bei der Realisierung eines ALM-basierten Verteildienstes für die Live-Übertragung multimedialer Daten ("Live-Streaming") sind neben den üblicherweise

an Übertragungsdienste für Multimedia-Daten gestellten Dienstgüteanforderungen nach einer möglichst geringen *Ende-zu-Ende-Verzögerung (Delay)* und *Schwankung dieser Verzögerung (Jitter)* auch Anforderungen in Bezug auf die Effizienz der Verteiltopologie zu beachten – letztere charakterisiert durch das *Verhältnis der Pfadlängen in der Topologie im Vergleich zum kürzesten Pfad ("Path Stretch")* und die *Anzahl von Kopien identischer Pakete auf einzelnen Teilstrecken ("Link Stress") [16].* Von ebenso großer Bedeutung für einen kommerziellen Einsatz dieses Ansatzes ist jedoch die *Gewährleistung einer hohen Verfügbarkeit* des Verteildienstes bei zufälligen Störungen sowie bei vorsätzlichen Sabotageangriffen.

In der Vergangenheit wurden mit unterschiedlichen Methoden bereits effiziente oder stabile Overlays konstruiert. Die bisherigen Verfahren zur Stabilisierung und Sicherung der Qualität übertragener Daten basieren hierbei in der Regel auf dem Hinzufügen von Redundanzen oder auf der Minimierung der Auswirkung einzelner ausfallender Teilnehmer auf das Gesamtsystem. Die erstgenannte Strategie nutzt die natürliche Robustheit multimedialer Daten gegen eine geringe Paketverlustrate und zielt darauf ab, die Auswirkungen, die einzelne ausfallende Knoten auf das restliche System haben, zu minimieren. Zu diesem Zweck wird zum einen versucht, durch die Konstruktion flacher Bäume die Anzahl der Vorgänger so gering wie möglich zu halten [2]. Um zum anderen weniger abhängig von einzelnen Vorgängern zu sein, versuchen weitere Ansätze den Datenstrom über möglichst unterschiedliche Pfade zu beziehen und dadurch den Knotenzusammenhang des Overlays zu erhöhen [5,13]. Beide Verfahren verhelfen potentiellen Angreifern jedoch zu dem Wissen über die Wichtigkeit von Knoten und versetzen sie so in die Lage besonders zentrale Knoten als Ziele auszuwählen. Insgesamt richten sich die vorgeschlagenen Ansätze damit lediglich gegen zufällige Ausfälle und nicht gegen vorsätzliche Angriffe.

Aufgrund der zeitlichen Anforderungen an den Transport von Multimediadaten und der Vermeidung von unnötigen Übertragungsvorgängen im Netzinneren (bezogen auf die Übertragungsvorgänge pro Netzwerk-Link) ist es für ALM weiterhin besonders wichtig, dass sich die Strukturen der konstruierten Topologien den Strukturen der unterliegenden Transportnetze anpassen. Ineffiziente Topologien, mit einer Vielzahl von Verbindungen zwischen weit entfernten Knoten führen nicht nur zu erhöhten Ende-zu-Ende-Verzögerungen und verstärktem Schwanken dieser Verzögerung (Jitter), sondern bewirken zugleich eine stärkere, unnötige Belastung des Transportnetzes. Die Konstruktion effizienter Topologien zielt daher darauf ab, Pakete vorrangig über kurze Ende-zu-Ende-Verbindungen zu übertragen und lange Distanzen im Netz zu vermeiden.

Um global möglichst gute Ergebnisse zu erreichen, verfolgen einige Ansätze [14] die Strategie, die gesamte Topologiekonstruktion einem Verwaltungsknoten zu überlassen. Der Ausfall eines solchen verwaltenden Knotens führt jedoch zu einem Zusammenbruch des Systems, so dass dieser Knoten ein gutes und allen Teilnehmern bekanntes Ziel für Angriffe darstellt.

Weitere Ansätze implementieren die Nachbarwahl verteilt und folgen hierbei grundsätzlich einer von drei Strategien: entweder wird mit Hilfe eines eigenen Signalisierungs-Overlays zunächst die eigene Position bestimmt und die Verbin-

dung zu einem der Knoten in der eigenen Region aufgebaut, oder der Knoten tritt dem Streaming-Overlay direkt bei und wird in diesem durch den lokalen Tausch mit Nachbarn sukzessive an eine kostengünstige Position gebracht [11,12]. In hybriden Verfahren sucht jeder Knoten zuerst einen nahe liegenden Nachbarn und nach dem Beitritt werden die lokalen Verbindungen optimiert [10]. Für die erste Strategie bedarf es eines Verfahrens, um die Positionen der einzelnen Teilnehmer initial zu ermitteln. Der zuverlässigste Ansatz hierfür ist es, eine Distanzbestimmung aller Knoten untereinander durchzuführen [7]. Er ist mit wachsender Teilnehmerzahl auf Grund seiner hohen Nachrichtenkomplexität jedoch nicht mehr durchzuführen. Eine vereinfachte Lösung des Verfahrens verfolgt lediglich das Ziel, Knoten der näheren eigenen Umgebung im Netzwerk zu identifizieren, diese zu gruppieren und daraufhin Verbindungen präferiert innerhalb der entstehenden Gruppen aufzubauen [15,1]. Diese Lösungen kommen mit einem signifikant geringeren Nachrichtenaufwand aus, der jedoch zu einer geringeren Qualität der Lösung führen kann.

Indirekt kann eine Reduktion der überbrückten Distanzen auch dadurch erreicht werden, dass der ALM auf einem Peer-to-Peer-Suchmechanismus, der die Lokalitäten der Knoten berücksichtigt, aufgesetzt wird und alle Pakete durch diesen geroutet werden [5,6]. Diese Lösungen beziehen jedoch in der Regel Knoten, die den empfangenen Datenstrom selbst nicht beziehen, zur Weiterleitung ein, was für diese nicht wünschenswert ist.

In unseren eigenen Vorarbeiten auf diesem Gebiet haben wir uns bisher vor allem auf die Konstruktion von Topologien konzentriert, die möglichst stabil gegen Angriffe sind. In [3] wurde eine Klasse von Topologien für einen Spezialfall (in Bezug auf Quellenkapazität und Teilnehmeranzahl) vorgestellt und die optimale Resistenz dieser Klasse gegen Sabotageangriffe bewiesen. Weiterhin wurde in [17] ein Verfahren für die verteilte und dynamische Konstruktion stabiler Streaming-Topologien vorgeschlagen und bewertet, das jedoch Effizienzkriterien noch nicht mit einbezieht.

Eine gemeinsame Betrachtung der Kriterien Effizienz und Stabilität gegen Sabotageangriffe von Topologien existiert unseres besten Wissens nach bisher nicht. Sie ist daher das Ziel des vorliegenden Artikels.

Die weiteren Abschnitte gliedern sich wie folgt: Abschnitt 2 beinhaltet eine formale Beschreibung der Konstruktionsaufgabe und der Bewertungsmetriken für Stabilität und Effizienz. In Abschnitt 3 stellen wir unseren Ansatz zur gemeinsamen Optimierung der beiden Kriterien vor. Hierzu werden zunächst Kostenfunktionen zur Abbildung von Stabilitäts- und Effizienzeigenschaften aufgestellt und darauf aufbauend ein verteilter Algorithmus für die dynamische Topologieoptimierung beschrieben. In Abschnitt 4 beschreiben und diskutieren wir die Ergebnisse einer Simulationsstudie des Ansatzes und Abschnitt 5 fasst die Ergebnisse des Artikels kurz zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

2 Stabilität und Effizienz in Streaming-Overlays

Grundlegend ist ein Overlay ein ungerichteter, schleifenfreier Graph $G = (V, E)$, bestehend aus einer endlichen Knotenmenge $V = \{v_1, \ldots, v_n\}$ (den teilnehmenden End-Systemen) und einer Menge Kanten $E \subseteq \{(u, v) | u, v \in V, u \neq v\}$ (den Verbindungen zwischen den End-Systemen).

Alle Endysteme sind in der Lage über das unterliegende Netzwerk paarweise miteinander Verbindungen aufzubauen, so dass der Graph $G = (V, E)$ im Allgemeinen zusammenhängend ist. Im konkreten Fall wählt jeder Knoten jedoch nur die Teilmenge $N_v \subseteq V$ als *Nachbarn*, wodurch in Overlays nur eine Teilmenge der Kanten vorkommt.

Da sich die Teilnehmer in Internets in paarweise unterschiedlichen Entfernungen zu einander befinden, ist zusätzlich die nichtnegative *Distanzfunktion*, die den Kosten der jeweiligen Kante entspricht, definiert: $\mathrm{d} : E \to \mathbb{R}^+$.

Zusätzlich sind die Netzzugänge der Teilnehmer in der Bandbreite beschränkt. Aus diesem Grund ist die *Knotenkapazität* wie folgt definiert: $\mathrm{c} : V \to \mathbb{R}^+$.

Im Overlay existiert eine Datenquelle $v_s \in V$, welche den Datenstrom mit der Bitrate R_0 erzeugt und als Ursprung den anderen Knoten zur Verfügung stellt. Dieser Datenstrom $\mathcal{S} = \{P_1, \ldots, P_n\}$ besteht aus n Datenpaketen, die auf jedem Knoten beliebig dupliziert und weitergeleitet werden können. Mittels Unterteilung des Datenstroms in l aufeinander folgende Sequenzen mit jeweils k Paketen und durch Zusammenfassung jedes i-ten Paketes aus allen Sequenzen kann er in k gleich große Teildatenströme, oder „*Stripes*", unterteilt werden:

$$\mathcal{S} = \{\{p_1^1, \ldots, p_k^1\}, \ldots, \{p_1^l, \ldots, p_k^l\}\} \text{ mit } p_j^i = P_{(i-1)\cdot k+j}$$

Mit C wird die Knotenkapazität der Quelle bezeichnet: $C = c(v_s)$, die Quelle hat in Konsequenz maximal $C \cdot k$ ausgehende Kanten und die Bandbreite des Netzzugangs der Quelle beträgt damit $C \cdot R_0$.

Im allgemeinen Fall, dass alle Knoten den gesamten Datenstrom erhalten, werden die Pakete des Datenstromes entlang n Spannbäumen T_i, Out-Trees mit der Wurzel v_s, über den Graphen verteilt. Im Weiteren wird von der Unterteilung des Datenstroms in Stripes ausgegangen. In diesem Fall gilt, bei Vernachlässigung der Knotendynamik, dass die jeweils l Spannbäume eines Stripes identisch sind und lediglich k unterschiedliche Spannbäume existieren: $T_1 = (V, E_1), \ldots, T_k = (V, E_k)$. Aufgrund der Bandbreitenbeschränkung der einzelnen Knoten gilt darüber hinaus, dass die Summe der Grade in den Spannbäumen T_i für jeden Knoten $v \in V$ höchstens $c(v)$ ist:

$$\sum_{i=1}^{k} \deg_{T_i}(v) \leq c(v) \quad \text{für alle } v \in V$$

Mit den Distanzen als Kosten berücksichtigt ergeben sich die Gesamtkosten der Topologie in diesem Fall zu:

$$\text{totalcost}(\mathcal{T}) = \sum_{i=1}^{k} d(T_i) = \sum_{i=1}^{k} \sum_{e \in E_i} d(e).$$

Die Sequenz $\mathcal{T} = (T_1, \ldots, T_k)$ der Spannbäume wird im Weiteren als Streaming-Topologie $\mathcal{T}$ bezeichnet.

Verlässt ein Knoten v das Overlay, so führt dies bis zur erneuten vollständigen Verbindung der Topologie zu einem Verlust der Pakete des Stripes i bei allen seinen Nachfolgern $\text{succ}_{T_i}(v)$. Die Anzahl der dadurch nicht mehr empfangenen Stripes beschreibt die Funktion $a_{T_i}(\mathcal{A}) := |\text{succ}_{T_i}(\mathcal{A}) \cup \mathcal{A}|$. Die Anzahl der durch eine Ausfallmenge $\mathcal{A}$ in der gesamten Topologie $\mathcal{T}$ verursachten Paketverluste, ergibt sich zu: $a_{\mathcal{T}}(\mathcal{A}) := \sum_{i=1}^{k} a_{T_i}(\mathcal{A})$.

Als Maß für die Robustheit eines Overlays wird damit die *Angriffsstabilität* der Topologie definiert. Sie beschreibt die Anzahl der Knoten, die aus einem Overlay entfernt werden müssen, um eine Schranke Θ_{drop} insgesamt im System verlorener Pakete zu überschreiten: Gegeben ist die Topologie $\mathcal{T}$ und eine Paketverlustschranke $\Theta_{Drop} \in (0, 1)$.
Gesucht ist die minimale Ausfallmenge $\mathcal{A} \subseteq \mathcal{T} \setminus \{v_s\}$ von Knoten, so dass gilt:

$$a_{\mathcal{T}}(\mathcal{A}) \geq \Theta_{Drop} \cdot k \cdot |V|.$$

Die Metriken Link-Stress und Path-Stretch für die Bewertung der Effizienz der konstruierten Topologien sind stark von der Struktur des unterliegenden Transportnetzes abhängig: In großen Infrastrukturnetzen, die aus vielen Netzzwischensystemen und Verbindungen zwischen diesen bestehen, überdecken sich die kürzesten Pfade zwischen unterschiedlichen Endsystemen seltener, als in kleineren Infrastrukturnetzen.

Zudem sind die direkten Netzwerkpfade zwischen der Datenquelle und vielen der Endsysteme größer als in kleinen Infrastrukturnetzen, während die Netzwerkpfade zwischen nahe liegenden Endsystemen (die etwa am gleichen Access-Router angeschlossen sind) gleich bleiben.

Diese Tatsachen führen dazu, dass sowohl der Link-Stress als auch der Path-Stretch der gleichen Overlay-Konstruktionsprozedur in großen Infrastrukturnetzen mit vielen Netzwerklinks geringer als in kleinen Infrastrukturnetzen sind.

Aus diesem Grund bedarf es einer Metrik, mit Hilfe derer exaktere und vergleichbare Aussagen über die Effizienz einer Konstruktionsprozedur getroffen werden können. An dieser Stelle wird die Anzahl der gegenüber einer optimalen Topologie zusätzlich auf Punkt-zu-Punkt-Verbindungen übertragenen Netzwerk-Pakete (Hops) als Maß für die Effizienz in Bezug auf die Netzwerkkosten vorgeschlagen.

Hierfür wird als Optimalitätsmaß eine rein theoretische Lösung ohne Berücksichtigung jedweder Bandbreitenbegrenzungen gewählt. Der so theoretisch minimale Wert lässt sich mit globalem Wissen über das gesamte Netzwerk über das Overlay-Modell berechnen: Der effizienz-optimale Verteilbaum, der die geringsten Netzwerkkosten erzeugt, entspricht dem minimalen Spannbaum (MST) über alle Knoten und kann als Referenzwert $\text{totalcost}(MST(G))$ für jedes Overlay in einem zugehörigen Infrastrukturnetzwerk angegeben werden.

Das Maß für die Netzwerkeffizienz einer Topologie ergibt sich zu:

$$\text{hop_penalty}(G, \mathcal{T}) := \frac{\text{totalcost}(\mathcal{T})}{\text{totalcost}(MST(G))}.$$

3 Ein Ansatz zur gemeinsamen Optimierung

Ein neu zum System hinzukommender Knoten verbindet sich initial mit anderen Knoten in der Topologie, von denen er daraufhin die Datenstrompakete erhält. Zur Verbesserung der Eigenschaften der so zufällig entstehenden Topologie wird diese daraufhin optimiert. Um eine Skalierbarkeit auf große Gruppen zu ermöglichen und einen einzelnen, allen bekannten, Single-Point-of-Failure zu vermeiden, wird die Optimierung auf allen Knoten, basierend auf lokalem Wissen, verteilt implementiert.

Stabilität und Effizienz der Topologien sollen durch die eine lokale Optimierung mit Hilfe einer Gesamtkostenfunktion erreicht werden. Zur Optimierung der Topologie analysiert jeder Knoten seine aktuelle Situation und versucht die auftretenden Kosten zu minimieren. Dabei werden die Kosten aller Kanten zu den Kindern analysiert und einzelne Kanten gegebenenfalls untereinander weitervermittelt. Alle Veränderungen der Topologie sind damit von Vaterknoten initiiert. Um ein Abwägen zwischen den Optimierungszielen zu ermöglichen, werden die Kosten einer Kante in Bezug auf die Effizienz über den Faktor s gewichtet mit den Stabilitätskosten zusammengefaßt. Für die Kante des Knotens v zu seinem Kind w im Spannbaum des Stripes i ergeben sich damit die Gesamtkosten zu:

$$K_i(v, w) = s \cdot K_{stabil}(v, w, i) + (1 - s) \cdot K_{distanz}(v, w).$$

Zur Kostenminimierung müssen die alternativen Situationen überprüft werden, um den besten alternativen Vater u und dadurch die lokale Veränderung zu ermitteln, welche zur stärksten Kostensenkung führt.

Um eine Stabilität der Topologien gegen Ausfälle zu erreichen, müssen die Abhängigkeiten zwischen den Knoten minimiert werden.

Zunächst ist es für jeden Knoten wichtig, von keinem anderen Knoten große Anteile des Datenstroms zu beziehen, damit dessen Ausfall nicht zu hohem Paketverlust führt. Um dieses Ziel zu erreichen, werden der Datenstrom an der Quelle in Stripes unterteilt und für jeden Knoten Kosten für die Weiterleitung mehrerer Stripes eingeführt.

$$K_{sel}(v, i) := 1 - \frac{\text{fanout}_{T_i}(v)}{c(v)}.$$

Hierbei beschreibt $fanout_{T_i}(v)$ die Anzahl der ausgehenden Kanten des Knotens v im Spannbaum i. Durch die Minimierung dieser Kosten wird im besten Fall von jedem Knoten nur ein Stripe weitergeleitet.

Global ist es für die Topologie wichtig, dass Knoten, die im betrachteten Spannbaum Daten weiterleiten können, der Quelle möglichst nahe sind, um

die verfügbare Bandbreite zu erhöhen, und die anderen Knoten als Blätter der Quelle möglichst weit entfernt sind. Aus diesem Grund erhalten Knoten w, die den Stripe i weiterleiten können, die Kosten $K_{forw}(w,i) = 0$, anderenfalls $K_{forw}(w,i) = 1$, was dazu führt, dass bei Minimierung der Kosten die Spannbäume flach gehalten werden.

Außerdem sollten Knoten w mit vielen Nachfolgern von einem Vater v nicht tiefer gehängt werden, um die durchschnittliche Tiefe der Knoten in den Bäumen und damit die Abhängigkeit nicht zu erhöhen. Für die Angriffsstabilität ist es zusätzlich wichtig, dass der Ausfall eines bestimmten Knotens nicht zu deutlich höheren Paketverlusten führt, als der eines beliebigen anderen. Wenn einzelne Knoten mit einer hohen Anzahl von Nachfolgern $a_{\mathcal{T}}(v)$ existieren, so wird ein Angreifer in die Lage versetzt, durch das Ausschalten dieser wenigen Knoten große Teile des Systems vom Dienst zu trennen und leicht großen Schaden anzurichten. Beide Ziele werden durch die Balancierung der Topologie erreicht und es ergibt sich die Kostenfunktion:

$$K_{bal}(v,w,i) := \frac{\left(\frac{\text{succ}_{T_i}(v)}{\text{fanout}_{T_i}(v)} - 1\right) - \text{succ}_{T_i}(w)}{\left(\frac{\text{succ}_{T_i}(v)}{\text{fanout}_{T_i}(v)} - 1\right)}$$

Aus diesen drei Kostenarten ergeben sich die Stabilitätskosten insgesamt zu:

$$K_{stabil}(v,w,i) = K_{sel}(v,i) + K_{forw}(w,i) + K_{bal}(v,w,i).$$

Zur Konstruktion zusätzlich netzwerkeffizienter Topologien ist es wichtig, dass alle Kanten in den Spannbäumen eine möglichst geringe Distanz $d(v,w)$ aufweisen, um die Datenstrompakete auf kurzen Wegen zu übertragen. Aus diesem Grund werden alle Distanzen in den Spannbäumen minimiert.

$$K_{distance}(v,w,u) := 1 - \frac{d(e_{(v,w)}) + d(e_{(w,u)})}{d(e_{(v,w)}) + d(e_{(v,u)})}.$$

Diese lokale Optimierung führt auch global für die Topologie $\mathcal{T}$ zu einem niedrigen $totalcost(\mathcal{T})$ und damit zu geringem Link-Stress und $hop_penalty(\mathcal{T})$.

Da ein Vater nicht alle lokalen Stabilitätsinformationen seiner Kinder und ihrer Nachfolger kennt, wird die Gesamtoptimierung in zwei Schritten durchgeführt. Als erstes wird die Kante ausgewählt, welche die höchsten Gesamtkosten $K_i(v,w)$ verursacht. In einem zweiten Schritt wird bei der Optimierung der lokalen Situation von den Stabilitätskosten lediglich K_{bal} berücksichtigt und insgesamt so optimiert, dass der Nutzen $G_i(v,w,u) = K_i(v,w) - (s \cdot K_{bal}(v,u,i) + (1-s) \cdot K_{distance}(v,w,u))$ maximiert wird. Da durch die Optimierung die Baumhöhe lediglich erhöht werden kann, was bei einer Senkung von Stress und *totalcost* zu einer Erhöhung des Path-Stretch führt, wird als Nutzenschranke Θ_{pass} eingeführt und Optimierungen nur dann ausgeführt, wenn ihre Nutzensteigerung G diese Schwelle überschreitet. Die Entscheidung über die Weiterleitung eines Knotens kann nicht über eine konstante Schwelle parametrisiert werden, da einerseits vermieden werden muß, dass ein Knoten, der noch gar keine Bandbreite

verbraucht hat, sofort alle Knoten, die eine Verbindung zu ihm aufbauen, weiterleitet. Andererseits muß ein Knoten mit ausgelasteter Bandbreite in der Lage sein, eines seiner Kinder weiterzuleiten. Die Schwelle muß folglich durch eine Funktion über die Bandbreite berechnet werden. Sie soll mit steigender anteiliger Bandbreitennutzung eines Vaterknotens $b \in (0,1)$ stetig von 1 auf 0 fallen, wobei: $b = \frac{\deg(v)}{c(v)}$. Zusätzlich soll mit dem Gewicht t eine Optimierung auf geringeren Stretch oder geringere *totalcost* ermöglicht werden. Hierfür wurde die Funktion $\Theta_{pass}(b) = (1 - b^{(2 \cdot t)^3} \cdot (1 - t^3)) + t^3$ gewählt, wobei der Funktionswert von $\Theta_{pass}(1)$ auf $-\infty$ definiert ist. Stellt ein Knoten fest, dass er Bandbreite frei

Algorithmus 1 Topologieoptimierung auf Knoten v

Input: v, N_v
$d \leftarrow \emptyset$ {zu entfernende Kante}; $a \leftarrow \emptyset$ {alternatives Parent}; $b \leftarrow deg(v)$
gain $\leftarrow$ *wahr*; i $\leftarrow$ präferierter Stripe
while gain **do**
  gain $\leftarrow$ *falsch*
  $d \leftarrow w$: $K(v, w, \text{Childs}_{T_i}(v), i) = \max\{K(v, w, \text{Childs}_{T_i}(v), i) \mid w \in \text{Childs}_{T_i}(v)\}$
  $a \leftarrow w$: $G(v, w, a, i) = \max\{G(v, w, a, i) \mid w \in \text{Childs}_{T_i}(v) \setminus \{d\}\}$
  if G(v,a,d,i)$\geq \Theta_{pass}$ **then**
    drop(d,a)
    gain $\leftarrow$ *wahr*
  end if
end while
while $b < c(v)$ **do**
  a $\leftarrow w = rand\{\text{Childs}_{T_i}\}$
  requestChild(a,Θ_{pass}, $(\frac{\text{Succ}_{T_i}(v)}{\text{fanout}_{T_i}(v)} - 1)$)
  b $\leftarrow b + 1$
end while
a $\leftarrow w$: $\text{Succ}_{T_i}(w) = max\{\text{Succ}_{T_i}(w) \mid w \in \text{Childs}_{T_i}(v)\}$
if $\text{Succ}_{T_i}(a) > \frac{\text{Succ}_{T_i}(v)}{2}$ **then**
  requestChild(a,Θ_{pass}, $(\frac{\text{Succ}_{T_i}(v)}{\text{fanout}_{T_i}(v)} - 1)$)
end if

hat, so fordert er von einem seiner Kinder einen beliebigen Nachfolger an. Hierdurch wird erreicht, dass bei freiwerdenden Bandbreiten die Höhe der Topologie gesenkt werden kann.

Mit den so definierten Funktionen für Kosten K und die Kostensenkung G optimiert der Algorithmus zur lokalen Topologieoptimierung (vgl. Algorithmus 1) die lokale Nachbarschaft jedes teilnehmenden Knotens. Die Zeitkomplexität dieser Optimierung ist in jedem Optimierungsschritt für jeden Knoten quadratisch in der Anzahl seiner Kinderknoten im optimierten Spannbaum.

4 Analyse und Simulationsstudie

Zur Überprüfung der Hypothese, dass mit Hilfe lokaler Optimierungen auf Basis der vorgestellten Kostenfunktionen stabile und effiziente Topologien konstruiert werden können, wurde eine zweiteilige Simulationsstudie durchgeführt.

Zu diesem Zweck wurden zunächst Algorithmus und Signalisierungsprozedur im Simulationswerkzeug OMNeT implementiert und so Topologien konstruiert.

Als Backbone kam ein durch den BRITE-Topologiegenerator konstruiertes Netzwerk, welches dem Internet-Modell aus [4] folgend generiert wurde, mit 750 Netzzwischensystemen zum Einsatz. Mit den Zwischensystemen wurden gleichverteilt eine Datenstromquelle und zwischen 50 und 250 End-Systeme verbunden, die einem Nutzermodell nach [8] folgend, dem System beitreten.

Um zur Erlangung der für die Effizienzoptimierung benötigten Entfernungen eine permanente Distanzmessung zwischen den Knoten zu vermeiden, wurde ein synthetisches Koordinatensystem basierend auf Vivaldi [9] implementiert. Als Metrik kommt in der Implementierung die Anzahl der Punkt-zu-Punkt-Verbindungen (IP-Hops) im Netzwerk zum Einsatz. Die Abweichung zwischen geschätzten und gemessenen Entfernungen zum Ende der Simulationszeit betrug dabei im Maximum ca. 218% und im Schnitt ca. 45%.

Zur Untersuchung wurde der Aufbau bei gegebenem Nutzermodell simuliert und die entstehenden Topologien danach offline analysiert.

Um die Effizienz der entstandenen Topologien zu bewerten, wurden die zur Übertragung des Streams benötigten Kosten *totalcost* und die Güte in Form des *hop_penalty* ermittelt.

Dabei wurde zunächst das Stabilitätsgewicht s auf 0 festgelegt und das Gewicht t variiert. Eine Erhöhung von t führte zu insgesamt flacheren Overlay-Bäumen und dadurch zu geringerem Path-Stretch, da die Nutzenschranke, die der Nutzen einer Optimierung erbringen muß, damit sie durchgeführt wird, steigt. Parallel zur Erhöhung des Stretch sinkt der Link-Stress, da mehr Knoten an der tatsächlichen Verteilung des Datenstroms beteiligt und die Daten über mehr unterschiedliche Netzwerklinks übertragen werden.

Bei einer Senkung von t konnte außerdem ein verringertes *hop_penalty* beobachtet werden. Auch diese Beobachtung war zu erwarten, da bereits Optimierungen mit geringerem Effizienznutzen durchgeführt wurden. Eine Grenze stellte sich bei einem Wert von 0.2 ein. Wurde der Parameter unter diesen Wert gesenkt, konnten die Effizienzeigenschaften nicht stetig verbessert werden. Diese Tatsache ist damit zu erklären, dass die Optimierungen bei geringeren Werten nur einen sehr kleinen Nutzen erbringen müssen. In der verteilten Optimierung führt dies dazu, dass im Aufbau der Topologie auch lokale Optimierungen, die sich global betrachtet als ungünstig erweisen, durchgeführt werden, und die Optimierung daraufhin nur noch lokale Minima erreicht. Auf Grund der guten Ergebnisse für die Effizienz wurde der Parameter für t daraufhin auf 0.2 festgelegt.

Insgesamt konnte der minimale Spannbaum bei steigenden Gruppengrößen immer weniger gut angenähert werden und das minimale *hop_penalty* stieg stetig mit wachsender Teilnehmerzahl. Diese Tatsache ist nicht überraschend, da die Kapazitäten der Knoten bei der realen Optimierung berücksichtigt werden müssen, während sie für den minimalen Spannbaum keine Rolle spielen.

Im zweiten Schritt sollte der Trade-Off zwischen der Effizienz und der Stabilität untersucht werden, wozu das Gewicht s variiert wurde.

Die Resultate dieser Untersuchung (vgl. Abb. 1) zeigten, daß die Topologien bei niedrigem Gewicht s erwartungsgemäß effizient wurden. Mit steigendem Gewicht und zunehmender Optimierung der Topologien auf Stabilität stieg diese Anzahl in allen Simulationen erwartungsgemäß stetig an. Für alle Gruppengrößen gilt hierbei, daß die Anzahl verschickter Netzwerkpakete immer geringer war, als in einem Client-Server-Szenario.

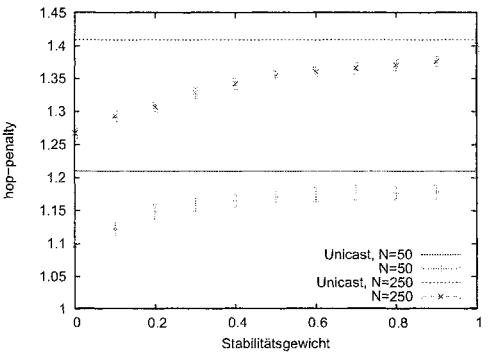


Abb. 1. Hop-Penalty entstehender Topologien im Vergleich zu Client-Server-Unicast-Streaming (16 Simulationsläufe, 98% Konfidenz)

Da die Optimierung stark von der Güte der Koordinaten abhängig ist, wurde das synthetische Koordinatensystem bei der Evaluation als Schwachstelle identifiziert. Die Dauer der Lernphase, die benötigt wurde, um die Distanzen zwischen Knoten zuverlässig genug vorhersagen zu können, lag um ein Weites höher als erwartet. Um eine Abweichung der Koordinatenschätzung unter 700% zu senken bedurfte es einer Vorlaufzeit von 300 Simulationssekunden, wobei der Nachrichtenaufwand in $\mathcal{O}(N)$ blieb.

Resultierend kann festgestellt werden, daß die Effizienzoptimierung den Erwartungen entspricht und die Topologien bei geringerer Gewichtung der Effizienz höhere Kosten im Netzwerk erzeugen.

Zur Untersuchung der Stabilitätseigenschaften der Topologien wurden sie zusätzlich auf ihre Angriffsstabilität untersucht. Zu diesem Zweck wurde ein auf globalem Wissen basierender Greedy-Angriff implementiert, der die Knoten nach der Anzahl ihrer Nachfolger aus der Topologie entfernt, und die daraus resultierende Paketverlustrate im Gesamtsystem gemessen. Die Angriffsstabilität gibt damit an, wieviele Knoten mindestens aus der Topologie entfernt werden müssen, um die insgesamt noch ausgelieferten Pakete auf eine vorgegebene Schwelle zu senken. Im Ergebnis zeigt sich somit der maximale durch korrelierten Knotenausfall auftretende Paketverlust im Gesamtsystem für den Zeitraum bis zur Reparatur der Topologie.

Diese zweite Auswertung zeigte, daß sich die Topologien bei einer reinen Stabilitätsoptimierung ($s = 1.0$) einer optimalen Topologie annäherten und die Paketverluste den Ergebnissen aus vorherigen Arbeiten entsprachen [3].

Eine Senkung von s führte wie erwartet zu einer Verringerung der Stabilität der Topologien (vgl. Abb. 2). Allerdings konnten starke Verringerungen in der Stabilität der Topologien erst ab Gewichtungen von $s < 0.2$ beobachtet werden.

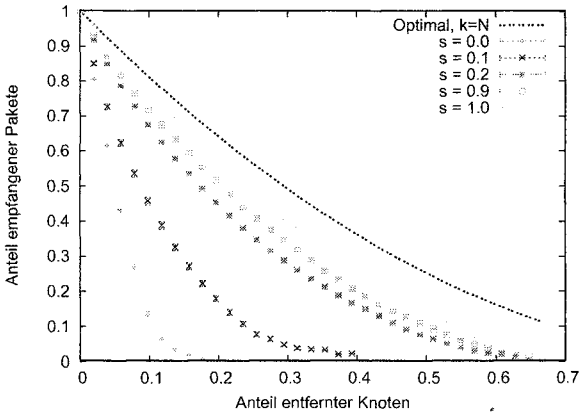


Abb. 2. Minimaler Anteil empfangener Pakete im System nach Anteil entfernter Knoten bei unterschiedlicher Stabilitätsgewichtung (16 Simulationsläufe, 98% Konfidenz)

Bereits mit geringer Gewichtung s führte die Optimierung folglich zu Topologien mit hoher Knotenkonnektivität und balancierten Bäumen.

Eine zusätzliche Untersuchung zeigte darüber hinaus, dass eine weitere Variation des Gewichtes t bei beliebigen s erwartungsgemäß zu keiner weiteren Effizienzsteigerung führte.

Zusammenfassend ist festzustellen, dass der Algorithmus mit einer Gewichtung von $s = 0.2$ und $t = 0.2$ zu sowohl angriffsstabilen als auch netzwerkeffizienten Topologien führt.

5 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein formales Modell für die Analyse von Overlay-Streaming-Systemen aufgestellt. Mit Hilfe des Modells wurde als neue Effizienzmetrik $hop_penalty(G, \mathcal{T})$, das Verhältnis der Distanzsummen von konstruierter Topologie $\mathcal{T}$ und einem minimalen Spannbaum in G, eingeführt.

Zur Konstruktion von Topologien, die einen guten Kompromiss zwischen Angriffsstabilität und Netzwerkeffizienz realisieren, wurde ein verteilter Algorithmus beschrieben, der durch die Verbindung der Effizienz- und Stabilitätskosten in einer gewichteten Summe eine Abwägung zwischen beiden Optimierungszielen ermöglicht.

In einer Simulationsstudie wurde gezeigt, dass der Algorithmus trotz einer Beschränkung auf lokales Wissen mit Hilfe der Kostenoptimierung in der Lage ist, stabile und effiziente Topologien zu konstruieren. Dabei stieg den Erwartungen entsprechend die Angriffsstabilität bei einer erhöhten Gewichtung der Stabilität und sank das *hop_penalty* der Topologien bei einer erhöhten Gewichtung der Effizienz. Als Gewicht für die Kostensumme wurde ein Bereich ermittelt,

in dem die Topologien sowohl gute Effizienz- als auch Stabilitätseigenschaften aufweisen. In allen Simulationsszenarien lag das *hop_penalty* dabei unter dem *hop_penalty* eines Client-Server-Unicast für die gleiche Knotengruppe.

Offene Fragen für zukünftige Arbeiten sind die Bewertung der Stabilität gegenüber zufälligen Ausfällen von Knoten, für die die Angriffsstabilität lediglich eine obere Schranke angibt, sowie die Analyse der Signalisierungsprozedur auf Schwachstellen gegenüber vorsätzlichen Angriffen.

Literatur

[1] Banerjee, S. ; Bhattacharjee, B. ; Kommareddy, C.: Scalable application layer multicast. In: *ACM Computer Communication Review*, 2002, S. 205–217

[2] Birrer, S. ; Lu, D. ; Bustamante, F.E. ; Qiao, Y. ; Dinda, P.: FatNemo: Building a resilient multi-source multicast fattree. In: *WCCD*, 2004, S. 182–196

[3] Brinkmeier, M. ; Schäfer, G. ; Strufe, T.: *A Class of Optimal Stable P2P-Topologies for Multimedia-Streaming.* 2007. – submitted to IEEE INFOCOM'07

[4] Bu, T. ; Towsley, D.: On distinguishing between Internet power law topology generators. In: *INFOCOM Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies* Bd. 2, 2002 (Proceedings), S. 638–647

[5] Castro, M. ; Druschel, P. ; Kermarrec, A. ; Nandi, A. ; Rowstron, A. ; Singh, A.: SplitStream: High-bandwidth content distribution in a cooperative environment. In: *Proceedings of (IPTPS'03)*, 2003, S. 298–313

[6] Castro, M. ; Druschel, P. ; Kermarrec, A.M. ; Rowstron, A.: Scribe: a large-scale and decentralized application-level multicast infrastructure. In: *IEEE JSAC* 20 (2002), Nr. 8, S. 1489 – 1499

[7] Chu, Y. H. ; Rao, S. G. ; Seshan, S. ; Zhang, H.: A Case for End System Multicast. In: *IEEE JSAC* 20 (2002), Oktober, Nr. 8, S. 1456–1471

[8] Costa, C. ; Cunha, I. ; Borges, A. ; Ramos, C. ; Rocha, M. ; Almeida, J. ; Ribeiro-Neto, B.: Analyzing client interactivity in streaming media. In: *Proceedings of World Wide Web*, 2004, S. 534–543

[9] Dabek, F. ; Cox, R. ; Kaashoek, F. ; Morris, R.: Vivaldi: A Decentralized Network Coordinate System. In: *Proceedings of the ACM SIGCOMM'04*, 2004

[10] Francis, P.: *Yoid: Extending the internet multicast architecture.* 2000

[11] Jannotti, J. ; Gifford, D. ; Johnson, K. ; Kaashoek, M. ; O'Toole, J.: Overcast: Reliable Multicasting with an Overlay Network. In: *Proceedings of the Symposium on Operating System Design and Implementation*, 2000, S. 197–212

[12] Li, Z. ; Mohapatra, P.: HostCast: a new Overlay Multicasting Protocol. In: *IEEE ICC*, 2003, S. 702 – 706

[13] Liang, J. ; Nahrstedt, K.: DagStream: Locality Aware and Failure Resilient Peer-to-Peer Streaming. In: *Proceedings of MMCN* Bd. 6071, 2006. – to appear

[14] Padmanabhan, V. ; Wang, H. ; Chou, P. ; Sripanidkulchai, K.: Distributing streaming media content using cooperative networking. In: *Proceedings of ACM/IEEE NOSSDAV*, 2002, S. 177–186

[15] Ratnasamy, S. ; Handley, M. ; Karp, R. ; Shenker, S.: Topologically-Aware Overlay Construction and Server Selection. In: *Proceedings of IEEE INFOCOM* Bd. 3, 2002, S. 1190 – 1199

[16] Shimbel, A.: Structural parameters of communication networks. In: *Bulletin of Mathematical Biophysics* 15 (1953), S. 501 – 507

[17] Strufe, T. ; Wildhagen, J. ; Schäfer, G.: Towards the construction of Attack Resistant and Efficient Overlay Streaming Topologies. In: *Proceedings of STM*, 06

Improving the Performance and Robustness of Kademlia-Based Overlay Networks

Andreas Binzenhöfer[1] and Holger Schnabel[1]

University of Würzburg
Institute of Computer Science
Germany
{binzenhoefer, schnabel}@informatik.uni-wuerzburg.de

Abstract Structured peer-to-peer (p2p) networks are highly distributed systems with a potential to support business applications. There are numerous different suggestions on how to implement such systems. However, before legal p2p systems can become mainstream they need to offer improved efficiency, robustness, and stability. While Chord is the most researched and best understood mechanism, the Kademlia algorithm is widely-used in deployed applications. There are still many open questions concerning the performance of the latter. In this paper we identify the main problems of Kademlia by large scale simulations and present modifications which help to avoid those problems. This way, we are able to significantly improve the performance and robustness of Kademlia-based applications, especially in times of churn and in unstable states. In particular, we show how to increase the stability of the overlay, make searches more efficient, and adapt the maintenance traffic to the current churn rate in a self-organizing way.

1 Introduction

There is both theoretical and practical evidence that p2p networks have a potential to support business applications. They are scalable to a large number of customers, robust against denial of service attacks, and do not suffer from a single point of failure. Skype [12], a p2p based VoIP application, e.g., serves millions of people every day. The main task of the underlying p2p network is to support efficient lookups for content stored in the overlay. The latest generation of p2p networks, the so called Distributed Hash Tables (DHTs), was especially designed to handle this task in a fast and scalable way. There are numerous different DHTs proposed in literature: CAN, Pastry, Chord, and Kademlia, to name just a few. All those algorithms do have in common that each participating peer gets a unique identifier using a hash function, while a distance metric is defined on these identifiers. In order to maintain the stability of the overlay each peer usually has a very good knowledge about its neighbors and some additional pointers to more distant peers used as shortcuts to guarantee fast lookups. In the research community Chord became the most studied algorithm in the last few years, which is possibly due to its easy to analyze ring structure. The scalability [10] [2], the behavior under churn [5] and the overlay stability of Chord [3] are well understood.

The majority of deployed overlay networks, however, make use of the Kademlia protocol [7]. It replaces the server in the latest eMule modifications and is used as a distributed tracker in the original BitTorrent as well as in the Azureus client [1]. The latter continuously attracts more than 800.000 simultaneous users world wide. Despite all this there are only few scientific papers evaluating the performance of the Kademlia algorithm. In [6] the performance of different DHT algorithms including Kademlia is evaluated and compared. Modifications to support heterogeneous peers are introduced in [4]. Finally in [11] an analysis of the lookup performance of Kad, the Kademlia-based DHT used in eMule, is given. The authors examine the impact of routing table accuracy on efficiency and consistency of the lookup operation and propose adequate improvements.

In order to understand the performance of Kademlia in greater detail, we implemented a detailed discrete event simulator in ANSI-C based on the algorithm given in the original paper [7]. In particular, we studied the search duration, the overlay stability and the required maintenance traffic. In this paper we present the insights gained during our simulations. We will describe the weak points we discovered and pinpoint their root causes. For each problem we will present an optimization, which eliminates the disadvantages and makes Kademlia a protocol more feasible for business applications.

The remainder of the paper is structured as follows: In Section 2, we recapitulate the main aspects of the original Kademlia algorithm. A brief description of our simulator and the corresponding user model is given in Section 3. The discovered problems, their causes, and the solutions are summarized in Section 4. Section 5 finally concludes the paper.

2 Standard Kademlia

Kademlia is a DHT-based p2p mechanism which is used to efficiently locate information in an overlay network. A hash table is a data structure that associates keys with values. A distributed hash table (DHT) assigns the responsibility of parts of the value range of the hash function, i.e. of the address space S, to different peers. In order to retrieve the data, DHTs apply sophisticated routing schemes, such as self-balancing binary search trees. Each peer stores contact information about other peers in order to route query messages.

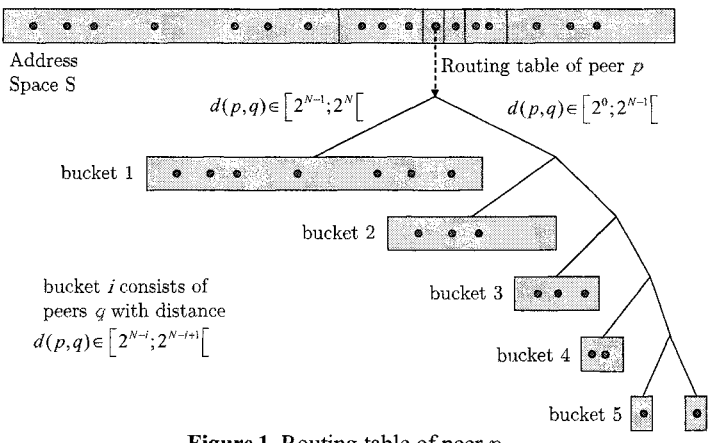


Figure 1. Routing table of peer p

In Kademlia, the branches of the binary search tree are represented as buckets, cf. Figure 1. The collection of buckets form the routing table. Bucket i of peer p's routing table is a list of peers which have a certain distance to peer p. Kademlia uses 160-bit identifiers for the address space and applies the XOR metric, i.e.,

$$S = \{0;1\}^N \text{ with } N = 160 \tag{1}$$

$$\begin{aligned} d: \quad S \times S &\rightarrow [0; 2^N], \\ (p, q) &\mapsto p \oplus q. \end{aligned} \tag{2}$$

This means that bucket i in the routing table of peer p covers all peers q with distance $d(p,q) \in [2^{N-i}; 2^{N-i+1}[$, cf. Figure 1. In order to keep the size of the routing table small enough, a bucket has at most k entries and is also referred to as k-bucket. This results in a maximal number of routing table entries of $k \cdot N$. A more detailed description of the Kademlia algorithm can be found in [7].

3 Simulator Details

In order to evaluate the different performance aspects of Kademlia, we developed a discrete event simulator according to the algorithms in [7]. As stated above, for each $0 \leq i < 160$ a peer keeps a bucket of k peers of distance between 2^{N-i} and 2^{N-i+1} from itself according to the XOR metric. Thereby the routing table is adapted dynamically. That is, each peer starts with one single bucket covering the entire address space and recursively splits the bucket containing the peer's own ID as soon as this bucket holds more than k entries. This results in an up-to-date routing table reflecting the current state of the overlay network as shown in Figure 1. When many peers leave the system, Kademlia merges the corresponding buckets accordingly.

Furthermore, a peer is able to insert documents into the overlay network. To guarantee their availability, each of these documents is stored at the k closest peers to the document's ID. If the document was not received from another peer for T_{rep} minutes, the corresponding peer republishes the document, i.e. it sends the document to the remaining $k-1$ peers of the replication group. When searching for a document a peer recursively sends parallel queries to the α closest peers it knows. The next recursion begins as soon as the peer received β answers. This guarantees that a searching peer will only run into a timeout if $\alpha - \beta + 1$ peers do not answer within one specific search step. If not stated otherwise, we use the default parameters $T_{rep} = 60$ minutes, $\alpha = 3, \beta = 2$, and $k = 20$.

To model end user behavior, we randomly chose join and leave events for each peer. To be comparable to other studies in literature a peer stays online and offline for an exponentially distributed time interval with a mean of E_{on} and E_{off} respectively. When online, the peer issues a search every E_{search} minutes, where the time between two searches is also exponentially distributed. Using different distributions mainly changes the quantitative but not the qualitative statements made during the remainder of this paper. To increase the credibility of our results [8], we include the 95 percent confidence intervals where appropriate.

4 Improvements

All structured p2p networks have been designed to scale to a large number of peers in the overlay. Therefore the real scalability issue of such systems is not in terms of system size but in terms of churn [9]. That is, the frequency at which peers join and leave the system has significantly more influence on its robustness and stability than the mere size of the system itself. In this section we uncover the problems caused by churn and show how to avoid them. In each simulation we use a total of 40000 peers, which we found to be sufficiently large to capture all important effects regarding the overlay size, and set $E_{on} = E_{off}$, resulting in an average overlay size of 20000 peers. The focus of our analysis of the simulation results is on qualitative behavior and not on quantitative statements.

4.1 Search Efficiency

The success and duration of a search for a document heavily depend on the correctness of a peer's pointers to other peers, i.e. on the correctness of the peer's routing table. In Kademlia the most crucial pointers are those to its k closest neighbors in the overlay. We measure the correctness of these pointers using two different variables:

- P_h: States how many of its current k closest neighbors a peer actually holds in its k-buckets.
- P_r: Represents the number of correct peers out of the k closest neighbors, which a peer actually returns when asked for.

Ideally a peer would not only know but also return all of its k neighbors.

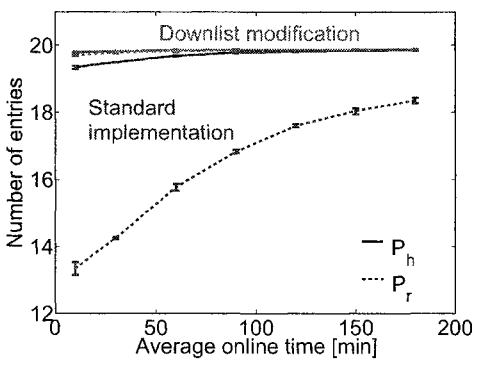


Figure 2. P_h and P_r in dependence of the churn rate

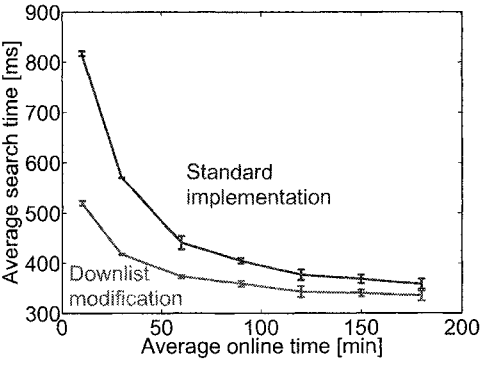


Figure 3. Influence of the downlist modification on the search efficiency

However, our simulations show that the standard implementation of Kademlia has problems with P_r. We set $k = 20$ and simulated the above described network for different churn rates. Figure 2 illustrates P_h and P_r in dependence of the churn rate. The mean

online/offline time of a peer was variied between 10 and 180 minutes. Even though on average a peer knows almost all its neighbors (P_h close to 20), it returns significantly less valid entries when queried (P_r as low as 13). The shorter a peer stays online on average, the less valid peers are returned during a search. The problem can be tracked down to the fact that there are still many pointers to offline peers in the corresponding k-bucket of the peer. The reason is that there is no effective mechanism to get rid of outdated k-bucket entries. Offline entries are only eliminated (or moved to the cache) if a peer runs into a timeout while trying to contact an offline peer. A peer which identifies an offline node, however, keeps that information to itself. Thus, it is not unlikely that a node returns offline contacts as it has very limited possibilities to detect offline nodes. As a result more timeouts occur and searches take longer than necessary. Another problem is that searches are also getting more inaccurate, which has negative effects not only on the success of a search but also on the redundancy of the stored documents. The reason is that due to the incorrect search results documents will be republished to less than k peers or to the wrong peers.

Solution - Downlists The primary reason for the above mentioned problem is that so far only searching peers are able to detect offline nodes. The main idea of our solution to this problem is that a searching peer, which discovers offline entries while performing a search, should share this information with appropriate other peers. To do so, a peer maintains a downlist consisting of all peers which it discovered to be offline during its last search. At the end of the search the corresponding entries of this downlist are sent to all peers which gave those entries to the searching peer during its search. These peers then also remove the received offline entries from their own k-buckets. This mechanism helps to get rid of offline entries by propagating locally gained information to where it is needed. With each search offline nodes will be eliminated.

The improved stability of the overlay is obviously bought by the additional bandwidth needed to send the downlists. From a logical point of view, however, it does require more overhead to keep the overlay stable under higher churn rates. In this sense, the additional overhead traffic caused by sending downlists is self-organizing as it automatically adapts to the current churn rate. The more churn there is in the system, the more downlists are sent.

It should also be mentioned, that without appropriate security arrangements a sophisticated attacker could misuse the downlist algorithm to exclude a target node by claiming in its downlist that this specific node had gone offline. However, this problem can be minimized by only removing those nodes which were actually given to the searching node during a search or additionally by verifying the offline status using a ping message. One could also apply trust or reputation based mechanism to exclude malicious nodes.

Effect on Search Efficiency To compare the downlist modification to the standard implementation we again simulated a scenario with 20000 peers on average and calculated the 95 percent confidence intervals. Figure 2 proves, that the downlist modification has the desired effect on P_r, the number of correctly returned neighbors. Using downlists

both P_h and P_r stay close to the desired value of 20, almost independent of the current churn rate. That is, even in times of high churn the stability of the overlay can be guaranteed.

This improved correctness of the overlay stability also has a positive influence on the search efficiency. In Figure 3 we plot the average duration of a search against the average online/offline time of a peer. In this context an overlay hop was modeled using an exponentially distributed random variable with a mean of 80 ms. Both curves show the same general behavior. The longer a peer stays online on average, the shorter is the duration of a search. However, especially in times of high churn, the downlist modification (lower curve) significantly outperforms the standard implementation. The main reason is that on average a peer runs into more timeouts using the standard implementation, as it queries more offline peers during a search. The effects on the maintenance overhead will be discussed in Section 4.3.

4.2 Overlay stability

When peers join and leave the overlay network, the neighbor pointers of a peer have to be updated accordingly. As mentioned above, the downlist modification greatly improves the correctness of the k closest neighbors of a peer. To understand this effect in more detail, we have a closer look at a single simulation run. We consider a mean online/offline time of 60 minutes and an average of 20000 peers for both the standard implementation and the downlist modification.

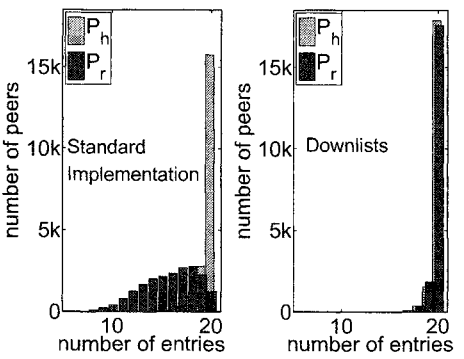


Figure 4. P_h and P_r for the standard implementation and the downlist modification

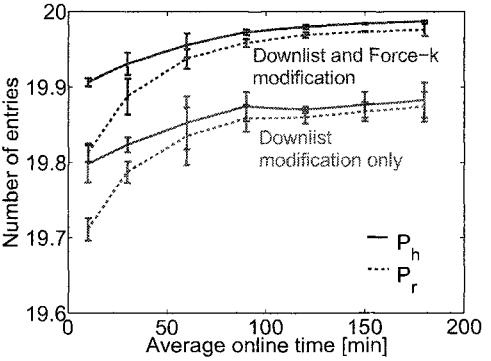


Figure 5. Effect of Force-k under churn

Figure 4 illustrates the distribution of P_h and P_r in both scenarios. As can be seen in the left part of the figure, almost all peers know more than 17 of their 20 closest neighbors using the standard implementation. However, the number of correctly returned peers P_r is significantly smaller for most peers. This problem is greatly reduced by the downlist modification as can be seen in the right part of the figure. In this case, the number of known and the number of returned peers are almost equal to each other. Yet, there are still some peers, which do not know all of their 20 closest neighbors. This

is in part due to the churn in the overlay network. However, simulations without churn produce results, which are comparable to those shown in the right part of Figure 4. The cause of this problem can be summarized as follows: Let B_p be the k-bucket of peer p, which includes the ID of peer p itself and $B_{\bar{p}}$ the brother of B_p in the binary tree whose leaves represent the k-buckets as shown in Figure 6. Then according to the Kademlia algorithm bucket B_p is the only bucket which will be split. However, if only $e < k$ of the actual k closest contacts fall into this bucket, then $v = k - e$ of these contacts theoretically belong into its brother $B_{\bar{p}}$.

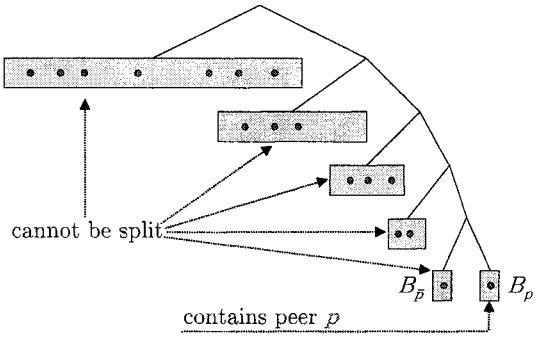


Figure 6. B_p and its brother $B_{\bar{p}}$ in the Kademlia routing table

Now, if this bucket is full it cannot be split. Thus, if some of the v contacts are not already in the bucket, it is very unlikely that the peer will insert them into its buckets. The reason is, that a new contact will be dropped in case the least recently seen entry of $B_{\bar{p}}$ responds to a ping message. Since in a scenario without churn all peers always answer to ping messages, new contacts will never be inserted into $B_{\bar{p}}$, even though they might be among the k closest neighbors of the peer. In the original paper it is suggested to split additional buckets in which the peer's own ID does not reside in order to avoid this problem. However, this has two major drawbacks. At first, it is a very complex process, which is vulnerable to implementation errors. Secondly, it involves a great deal of additional overhead caused by bucket refreshes and so on and so forth. In the next section, we therefore develop a simple solution, which does not require any additional overhead.

Solution - Force-k As stated above, it is possible, that a peer does not know all of its k closest neighbors, even in times of no churn. To solve this problem, we need to find a way to force a peer to always accept peers belonging into $B_{\bar{p}}$ in case they are amongst its k closest neighbors. Suppose a node receives a new contact, which is among its k closest neighbors and which fits into the already full bucket $B_{\bar{p}}$. So far, the new contact would have been dropped in case the least recently seen entry of $B_{\bar{p}}$ responded to a ping message. Compared to this, the Force-k modification ensures that such a contact will automatically be inserted into the bucket. In order to decide which of the old contacts will be replaced, one could keep sending ping messages and remove the first peer, which does not respond. However, this again involves additional overhead in terms of bandwidth. A faster and passive way is to put all entries of $B_{\bar{p}}$, which are not

among the k closest peers into a list l and drop the peer which is the least useful. This could be the peer which is most likely to be offline or the peer which has the greatest distance according to the XOR metric.

In our implementation, we decided to consider a mixture of both factors. Each of the entries e of list l is assigned a specific score

$$s_e = t_e + d_e \tag{3}$$

and the one with the highest score will be dropped. Thereby, t_e is intended to be a measure for the likelihood of peer e to be offline and d_e for the distance of peer e to peer p. The exact values of t_e and d_e are obtained by taking the index of the position of the corresponding peer in the list, as if it was sorted ascending by the time least recently seen or by the peer's distance respectively. That is, if e is the least recently seen peer ($t_e = 1$) and has the third closest distance to peer p ($d_e = 3$) it is assigned a score of $s_e = 4$.

Effect on Stability We investigated the impact of the Force-k modification on the stability of the overlay network in various simulations. In scenarios without churn, all peers finally know and return all of their k closest neighbors. The corresponding figures show lines parallel to the x-axis at a value of $k = 20$. It is therefore more interesting to regard the overlay stability during churn phases.

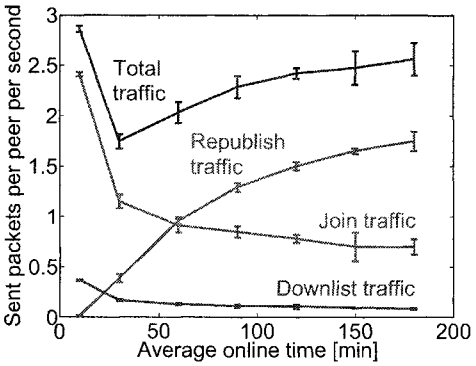


Figure 7. The maintenance traffic of a peer split into its components

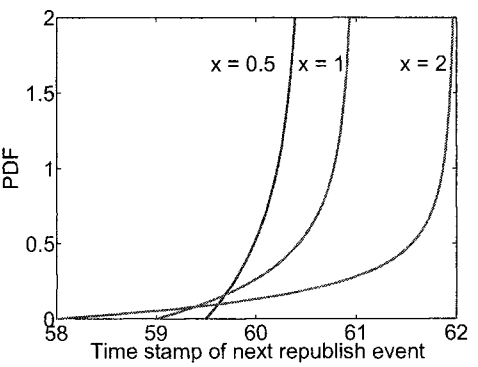


Figure 8. PDF of I_{rep} for different values of x

In Figure 5, we plot the average online time of a peer against the number of known and returned neighbors using the same simulation scenario as before. The two lower curves correspond to our previous results using the downlist modification. The two upper curves represent the Force-k modification in combination with the downlist modification. It can be seen that the Force-k algorithm also improves the stability of the overlay in times of churn. While the appearance of the curves is similar, there are more neighbors known (solid lines) and returned (dashed lines) as compared to using only the downlist modification. Even if a peer stays online for only 10 minutes on average,

it will know about 19.9 out of 20 neighbors and return more than 19.8 correct entries. By improving the correctness of the neighbors, the Force-k modification also increases the search success rate and the redundancy of stored documents.

4.3 Redundancy Overhead

The bandwidth required to maintain a stable overlay and to ensure the persistence of stored documents directly reflects the costs for a peer to participate in the network. We simulated a network with 20000 peers on average and recorded the average number of packets per second sent by a peer while it was online. Figure 7 illustrates the average traffic per peer in dependence of the average online time of a peer. In addition to the total traffic, the figure also shows its three main components, the join, the republish, and the downlist traffic.

Since E_{search}, the average time between two searches of a peer, was set to 15 minutes, the search traffic per peer per second can be neglected in this scenario and is thus not shown in the figure. The same is true for the traffic caused by bucket refreshes, since a specific bucket is only refreshed if it has not been used for an entire hour. The Force-k algorithm is performed locally and does thus also not produce any additional overhead.

It can be seen in the figure that the downlist traffic automatically adapts itself to the current churn rate. The more frequently the peers join and leave the system, the more downlist traffic is produced by a peer on average. In general, the small amount of bandwidth needed to distribute the downlists is also easily compensated by the improved stability of the overlay. The major part of the traffic is caused when joining the network and republishing documents. It is obvious that the average amount of join traffic increases if a peer stays online for a shorter period of time. The join traffic cannot and should not be avoided as it is necessary for a peer to make itself known when it joins the network. Moreover, the join traffic already shows a self-organizing behavior. The more churn there is in the system, the more joins there are in total and the more overhead is produced to compensate the problems caused by the churn.

At first, the run of the curve representing the republish traffic seems to be counter-intuitive. The less churn there is in the system, the more republish traffic is sent by a peer on average. However, the reason becomes obvious, if one takes into account that the longer a peer stays online on average, the more likely it gets that there are republish events. In fact, the probability that a peer stays online for longer than 60 minutes given the corresponding average online time E_{on}, resembles the run of the republish curve. The reason why the total amount of republish traffic exceeds the remaining traffic so significantly is as follows: Each document is stored at the k closest nodes to its ID, the so called replication group. To compensate for nodes leaving the network, each peer sends the document to all other peers of the replication group if it has not received the document from any other peer for $T_{rep} = 60$ minutes. The idea behind this republish mechanism is that one peer republishes the document and all other peers reset their republish timers accordingly. Since the republishing peer sends the document to all peers of the replication group simultaneously, the peers reset their timers at approximately the same time. The next time the first peer starts to republish the document, it has to search for the corresponding replication group before it can redistribute the document. However, during this search the republish timers of the other peers are likely to run

out and they will start to republish the document as well. For this reason, a document might get republished by up to k peers instead of just one single peer, resulting in unnecessary overhead traffic. This problem of synchronization is already mentioned in the original paper. In the following section, we present a solution, which greatly reduces the republish overhead and which is also resistant against churn.

Solution - Betarepublish The synchronization problem of the republish process arises if all peers of a replication group have approximately the same time stamp for the next republish event. At first this seems to be unlikely. However, each time a peer republishes a document all other peers of the replication group receive this document at approximately the same time and are thus synchronized again. The main idea to avoid this problem is to assure that all peers use different time stamps. To achieve this, each peer chooses its time stamp randomly in the interval $[T_{rep} - x, T_{rep} + x]$ instead of exactly after $T_{rep} = 60$ minutes. Let I_{rep} be the random variable describing the time stamp of the next republish event. Then we want I_{rep} to be distributed in such a way, that only few peers start republishing at the beginning of the interval and the probability to republish increases towards the end of the interval. This can, e.g., be achieved by setting:

$$I_{rep} = (T_{rep} - x) + 2 \cdot x \cdot I_{beta} \tag{4}$$

where I_{beta} is a random variable with density

$$i_{beta}(t) = \begin{cases} \frac{t}{\sqrt{(1-t) \cdot B(2,0.5)}} & \text{if } 0 < t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and $B(\alpha, \beta)$ is the beta function, defined by

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1} (1-t)^{\beta-1} \, \mathrm{d}t \tag{6}$$

Thereby x should be small compared to T_{rep} but still significantly larger than the duration of a search. Figure 8 shows the probability density function of I_{rep} for different values of x. All peers will set their time stamps somewhere in the interval $[60 - x, 60 + x]$. The probability for a peer to set its time stamp is still very low at the beginning of the interval. It then ascends significantly towards the end of the interval. In the case of $T_{rep} = 60$ minutes, $x = 2$ minutes is a reasonable choice, since it offers a long period of time with a low probability of republish events. This way, the republish traffic will be significantly reduced as it becomes very likely that only one or a few peers actually start a republish process. Again, note that a peer does only republish a document if it has not received it from another peer for $T_{rep} = 60$ minutes.

Effect on Overhead In this section we will have a look at the influence of the Betarepublish modification on the average amount of republish traffic sent by a peer.

Figure 9 shows the average number of republish packets per peer per second in dependence of the average online time. We compare the results for simulations using

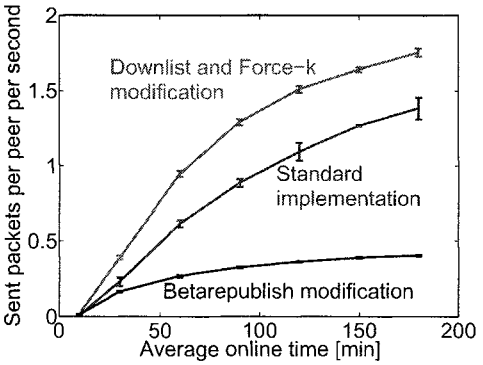


Figure 9. Maintenance traffic caused by republish processes

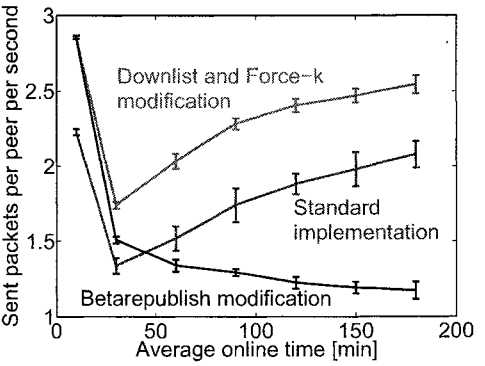


Figure 10. Total maintenance traffic in dependence of the churn rate

the standard implementation, our two previous modifications, and all modifications including Betarepublish. First of all the average republish traffic of a peer is increased by using the downlist modification. The reason is that using the standard implementation there are more offline nodes in the k-buckets during times of churn. Thus documents are republished to less peers, which reduces the republish traffic but also the redundancy in the system. The additional traffic introduced by the downlist modification is therefore used to improve the availability of documents.

The Betarepublish modification is applied in an effort to minimize the traffic which is necessary to achieve this availability. The figure shows that Betarepublish indeed reduces the amount of required republish traffic significantly. The Betarepublish traffic lies well beneath the standard implementation and also rises slower with an increasing average online time. Note that the Betarepublish modification does only avoid redundant traffic. It is still able to guarantee the same redundancy, stability, and functionality. Figure 10 shows how the reduced republish traffic influences the total traffic for the three regarded versions of Kademlia (Standard, downlists and Force-k, all modifications). At first, it can be seen that the use of downlists increases the total traffic as compared to the standard implementation. Again, this is desired overhead as it greatly helps to increase the robustness, the stability, and the redundancy of the overlay in an autonomous way.

By adding the Betarepublish modification, the total traffic is significantly reduced and no longer dominated by the republish traffic. While the average maintenance traffic sent by a peer in the standard implementation actually increases when there is less movement in the overlay network, it finally shows a self-organizing behavior when using all modifications. The less churn there is in the system, the less maintenance traffic is generated to keep the overlay network up to date. That is, the amount of bandwidth invested to keep the overlay running automatically adapts itself to the current conditions in the overlay.

5 Conclusion

In this paper we investigated the performance of the Kademlia protocol using a detailed discrete event simulator. We were able to detect and pinpoint some weak points regarding the stability and the efficiency of the overlay network. In this context, three modifications have been proposed to enhance the performance, the redundancy, and the robustness of Kademlia-based networks. With the help of downlists, the correctness of the neighbor pointers and the duration of a search is greatly improved. The Force-k modification ensures that a peer has a very good knowledge of its direct neighborhood, which greatly increases the stability as well as the overall performance. We also introduced a new republish algorithm, which significantly reduces the total traffic needed to keep the overlay running. The improved version of Kademlia shows a self-organizing behavior as the amount of generated maintenance traffic autonomously adapts to the current churn rate in the system.

The proposed modifications can be used to support large scale p2p applications, which are able to sustain dynamic user behavior. Even though the algorithms have been introduced using Kademlia, they are by no means restricted to this protocol. Especially the downlist and the Betarepublish mechanisms can easily be applied to other DHTs like Pastry, CAN, or Chord.

Acknowledgements

The authors would like to thank Robert Henjes, Tobias Hoßfeld, and Phuoc Tran-Gia for the insightful discussions as well as the reviewers for their valuable suggestions.

References

1. Azureus. URL: http://azureus.sourceforge.net/.
2. A. Binzenhöfer and P. Tran-Gia. Delay Analysis of a Chord-based Peer-to-Peer File-Sharing System. In *ATNAC 2004*, Sydney, Australia, December 2004.
3. Andreas Binzenhöfer, Dirk Staehle, and Robert Henjes. On the Stability of Chord-based P2P Systems. In *GLOBECOM 2005*, page 5, St. Louis, MO, USA, November 2005.
4. Youki Kadobayashi. Achieving Heterogeneity and Fairness in Kademlia. In *Proceedings of IEEE/IPSJ International Workshop on Peer-to-Peer Internetworking co-located with Symposium on Applications and the Internet (SAINT2004)*, pages 546–551, January 2004.
5. Supriya Krishnamurthy, Sameh El-Ansary, Erik Aurell, and Seif Haridi. A Statistical Theory of Chord under Churn. In *4th International Workshop on Peer-To-Peer Systems*, Ithaca, New York, USA, February 2005.
6. Jinyang Li, Jeremy Stribling, Thomer M. Gil, Robert Morris, and M. Frans Kaashoek. Comparing the performance of distributed hash tables under churn. In *Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS04)*, San Diego, CA, February 2004.
7. Petar Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *IPTPS 2002*, Cambridge, MA, USA, March 2002.
8. K. Pawlikowski, H.-D.J. Jeong, and J.-S. Ruth Lee. On credibility of simulation studies of telecommunication networks. In *IEEE Communications Magazine*, January 2002.
9. Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling Churn in a DHT. In *2004 USENIX Annual Technical Conference*, Boston, MA, June 2004.
10. Ion Stoica, Robert Morris, David Karger, M. Frans. Kaashoek, and Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *ACM SIGCOMM 2001*, San Diego, CA, August 2001.
11. Daniel Stutzbach and Reza Rejaie. Improving lookup performance over a widely-deployed dht. In *IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.
12. Skype Technologies. Skype. URL: http://www.skype.com.

Improved Locality-Aware Grouping in Overlay Networks

Matthias Scheidegger and Torsten Braun

IAM, Universität Bern, Neubrückstrasse 10, 3012 Bern, Switzerland

Abstract The performance of peer-to-peer and overlay networks depends to a great extent on their awareness of the underlying network's properties. Several schemes for estimating end-to-end network distances have been proposed to simplify this task. The mOverlay framework identifies groups of nodes that are near to each other in the network topology. Instead of distances between nodes mOverlay measures distances between groups. However, mOverlay's locating procedure has a number of drawbacks. We propose an alternate method for identifying groups using Meridian's closest node search. Simulation results based on PlanetLab measurements indicate that the Meridian-based approach is able to outperform mOverlay in terms of joining delay, the size of the identified groups, and their suitability for a distance estimation service.

1 Introduction

Peer-to-peer and overlay networks use logical topologies rather than the physical topology of the underlying network. This allows them to achieve many desirable aims like high scalability or high resilience to node failure. Nevertheless, creating a logical topology without considering the properties of the underlying network may lead to considerable inefficiency. A single logical link connecting two nodes may in fact span many links on the physical network. This property is commonly referred to as the stretch of an overlay topology. Overlay networks usually perform better if neighbors in the overlay topology are also close to each other in the physical network.

Most early designs for peer-to-peer and overlay networks, like the unstructured peer-to-peer networks Gnutella [7] and Freenet [8], are unaware of the underlying network. More recent designs often consider the properties of the underlying network in some way. For example, a binning scheme based on round-trip times [10] can be used to optimize content-addressable networks (CANs) [9]. The routing algorithm in Pastry [11] also considers the round-trip time between overlay nodes to find the optimal next hop when forwarding queries. Application-level multicast is especially sensitive to the choice of topology. Accordingly, these systems focus on optimizing this aspect. Scribe [13] uses Pastry's routing algorithm to build efficient multicast trees. The MULTI+ approach [12] hierarchically groups overlay nodes according to their IP network prefixes to create a multicast tree. This is based on the idea that similar IP prefixes indicate a neighborship in the network.

Since adaptation to network properties is a common problem in peer-to-peer and overlay networks, several frameworks and services have been proposed to help with this task. One of them is mOverlay [1], a locality-aware overlay that assigns nodes to groups depending on their distance in the underlying network. A dynamic landmark procedure determines the closest group for each overlay node. Each mOverlay group has a set of leader nodes, which store measured distances to other groups. Using this structure, overlay applications can easily distinguish between local links and more expensive links to other groups. Furthermore, these groups can be used as equivalence classes for distance estimation. If two nodes are in the same group, their respective distances to a remote node should be approximately the same. The distance between two nodes can therefore be estimated by the distance between their respective groups. In many cases an application only needs to find the closest overlay node to a given IP address and is not interested in the exact distance between the overlay node and the end system in question. Meridian [2] is a lightweight framework that efficiently solves this closest node problem. Each Meridian node keeps track of an incomplete, fixed-size set of other Meridian nodes, which is updated using a gossip protocol. When a node receives a closest node request it will choose the nearest node to the destination from its set and forward the request to that node.

In recent work [4] we have proposed an overlay distance measurement service using local groups similar to those in mOverlay. In addition, our approach is also able to detect if remote nodes are close together. This is achieved by analyzing time series of distance measurements to remote hosts (obtained using `ping` for example). Similarities in any two time series indicate that the respective remote nodes are close to each other. This enables two improvements. First, the service can be deployed locally because remote hosts only need to respond to standard tools like `ping` and do not have to run any special software. Second, when looking at the network from a given position, far away groups are often indistinguishable from each other and can be viewed as a single entity. In such cases we only need to store a single data record for a set of groups, which improves the scalability of the service. Initially, we planned to use mOverlay as a mechanism to identify local groups. However, we have found that replacing a part of mOverlay's locating algorithm with Meridian's closest node search improves its accuracy and reduces the time it takes for a node to join the overlay network. Our contribution in this paper is the modified locating algorithm, which we compare to the dynamic landmark procedure originally proposed for mOverlay in [1]. The resulting group structure is also useful to simplify the task of optimizing overlay topologies, which is a hard problem if the topology is large. Topologies based on groups preserve inter-node distances but are much smaller than topologies based on individual nodes and are thus well-suited for solving optimal routing problems.

The remainder of the paper is organized as follows: Section 2 discusses related work, including mOverlay and Meridian. Section 3 describes the modified locating algorithm. In Section 4 we present the simulators used to compare the two approaches, and we discuss the simulation scenarios and results in Section 5. Section 6 concludes the paper.

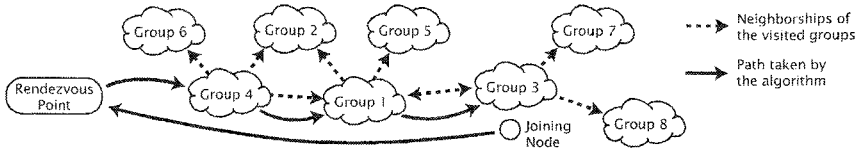


Figure 1. A joining node contacts the mOverlay rendezvous point and finds the nearest group 3 via groups 4 and 1.

2 Related Work

2.1 mOverlay

The mOverlay [1] framework uses a two tier overlay structure. At tier one, nodes that are close to each other form groups and communicate directly with other members of the same group. At tier two, groups select a number of nearby groups as their neighbors. The groups are chosen such that they can be used as equivalence classes concerning the distance metric. This reduces the endpoint-to-endpoint distance estimation problem to the much smaller problem of estimating distances between groups. Moreover, this structure can serve as a basis for constructing efficient overlay topologies because it distinguishes between efficient short distance links inside the groups and potentially inefficient long distance links between groups. In order to decide whether or not a joining node belongs to a given group the following *grouping criterion* is used [1]:

> When the distance between a new host Q and group A's neighbor groups is the same as the distance between group A and group A's neighbor groups, then host Q should belong to group A.

New nodes iteratively search for a group that meets this grouping criterion. When a node joins the overlay network it first contacts a rendezvous point and obtains contact information for a set of randomly chosen boot hosts. For each boot host it starts a locating process, which tries to find a suitable group for the node. Using several locating processes increases the robustness of the approach. The algorithm starts by contacting a boot host, which returns a set of distances between the boot host's own group and its neighbors. The joining node then measures and compares its own distances to these neighbor groups. If the grouping criterion is met the process terminates and the node joins the group of the boot host. Otherwise, the new node chooses the neighbor group that is nearest to it and repeats the process. After a predefined number of unsuccessful iterations, or if all available groups have been visited, the new node creates its own group. When a node creates a new group it selects its neighbors from the closest groups it has seen during the locating process, and their neighbors. It then contacts each of the selected neighbors in order to allow them to adjust their own neighbor tables if needed. Figure 1 illustrates the locating algorithm. Solid arrows indicate steps in the algorithm and dashed ones indicate neighborships between the groups that are used to check the grouping criterion.

2.2 Meridian

Meridian [2] is a "framework for performing network positioning without embedding nodes into a global virtual coordinate space." It has a another focus than mOverlay. Its three main functions are closest node discovery, central leader election, and multi-constraint search. For our work we use Meridian's closest node discovery. Meridian nodes form a loosely connected overlay network. They exchange information about other overlay nodes using a gossip protocol and keep track of a fixed number of peer nodes. These nodes are sorted into non-overlapping, concentric rings of exponentially growing width around the Meridian node (see Figure 2). The ith ring contains nodes with latencies between αs^{i-1} and αs^i from the center, and the outermost ring contains nodes with latencies αs^{i^*} and more. Within each ring, the nodes are selected to maximize diversity, which is quantified through the hyper-volume of the k-polytope formed by the selected nodes. A closest node search aims to identify the Meridian node that is

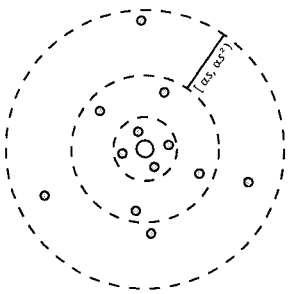


Figure 2. Meridian nodes arranged into rings based on their distance

closest to a given end system E in the network. To start the procedure we send a request to an arbitrary Meridian node. This node measures its latency to E and selects the nodes with similar latencies from its cache. It then contacts each of these nodes and asks them to measure and report their latency to E. The node with the smallest latency to E becomes the next hop, and the procedure repeats. When the next hop is only insignificantly closer than the current one the closest node search terminates and the current node is selected.

2.3 Other Work on Distance Estimation

A considerable amount of work on network distance estimation has been published in recent years. One of the earliest designs, IDMaps [14], is a distance estimation service that relies on tracers placed at key locations throughout the network. The distance between two clients is estimated by the sum of the distances between the nodes and their respective nearest tracers, plus the distance

between those tracers. Dynamic Distance Maps [15] uses a similar way to estimate distances, but uses the tracers to hierarchically decompose the Internet into regions. An important part of the work on network distance estimation focuses on coordinate-based approaches, which normally embed measured network distances in n-dimensional Euclidean space such that the Euclidean distance between two nodes is a good estimate of their distance in the network. GNP [16] is a prominent member of this family. Clients measure their distance to a fixed set of landmark nodes with known coordinates and compute their own coordinates using simplex downhill minimization. It has been argued that its fixed set of landmarks impairs GNP's scalability and makes it vulnerable to attacks and node failures. Consequently, more robust approaches like Lighthouse [17] have been proposed. Here, subset of overlay nodes may be used as landmarks, or lighthouses. Vivaldi [18] does not use any landmarks. Instead, it passively monitors network traffic to obtain distance measurements and applies a distributed algorithm to iteratively adjust the coordinates of the nodes.

3 An Alternative Locating Algorithm

A problem of mOverlay is its topology. Because the groups choose neighbors from their close proximity the logical links between the groups are very short. This affects the performance of the locating algorithm, since the algorithm follows the topology and thus can only make small steps towards the target node. If the target node is far away, taking bigger steps would be more efficient. Another problem is that mOverlay's topology is prone to so-called net-splits.

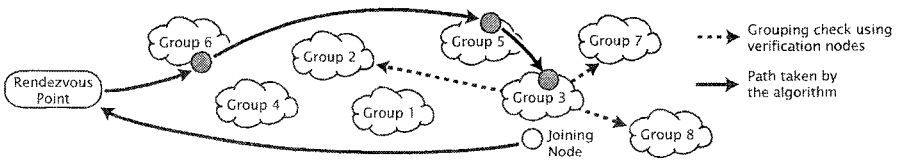


Figure 3. Alternative algorithm using Meridian's closest node search and mOverlay's grouping criterion

In order to overcome these problems we have defined an alternative group locating algorithm based on both mOverlay and Meridian. We take the group concept from mOverlay but change the overlay structure. The groups no longer have neighbors. Instead, the group leaders become Meridian nodes. When a new node wants to join, it locates a boot node using the rendezvous point. Then, it asks this boot node to start a Meridian closest node search with the joining node itself as target. The search returns the address of the closest group leader. At this point, the new node checks the grouping criterion to find out whether or not to join this group. If the criterion is met the new node joins the group. Otherwise, it creates a new group and becomes a Meridian node itself.

Unfortunately, we cannot directly use mOverlay's grouping criterion, because in our alternative approach groups do not have neighbors. We solve this problem using Meridian's node cache. The group leader found by Meridian's closest node search selects a randomly chosen set of *verification nodes* from its node table and creates a list of addresses and latencies to these nodes. The new node receives this list and in turn measures its latency to each of the verification nodes. This provides us with two comparable sets. Furthermore, because mOverlay's grouping criterion is formulated in general terms we also need to specify exactly when two distances can be considered "the same." We say distances x and y are the same if

$$\begin{array}{rcl} x \geq y & \wedge & (1-g) \cdot x \leq y \\ \vee\, x < y & \wedge & (1-g) \cdot y \leq x \end{array} \tag{1}$$

for a *grouping threshold* $g \in [0, 1)$. The test checks the relative difference between two distances. For example, with a grouping threshold of 0.05 we consider two distances the same if they are within $\pm 5\%$ of each other. A new node joins a group if test (1) succeeds for every verification node. The algorithm is illustrated in Figure 3.

We believe that this combined approach to grouping nodes solves the problems discussed above. Meridian's closest node search makes the search more efficient. The approach is also less prone to net-splits than mOverlay because Meridian nodes maintain a more diverse set of peer nodes. The loose overlay structure also makes the system more resilient to node failure. A possible drawback of our algorithm is that it only checks the grouping criterion for a single group, which bears the danger that the algorithm might skip over the optimal one. Fortunately, the results in Section 5 suggest that this kind of error is rare.

4 Implementation of Simulations

In order to compare the performance of mOverlay's locating algorithm to our alternative algorithm we have implemented simulators for the two approaches. Both simulators are based on a black box network model given by a matrix of the end-to-end latencies between each pair of endpoints in the simulation. For our experiments we use a matrix derived from all-sites ping data measured on PlanetLab [5]. In both simulators the nodes join the overlay network one after the other, in pseudo-random order (given by the seed value). For each node we record the time that expires until it joins a group or creates its own group. When the simulation ends we examine the resulting groups according to several criteria, which we discuss in Section 5.

4.1 mOverlay

We simulate mOverlay with a simple message-based approach where each message fits into a single packet and the message processing at a node does not take any time. Thus, a request-response message exchange takes exactly one round-trip time to complete, which is a lower bound for any real implementation of the

framework. Furthermore, we skip mOverlay's initial request to the rendezvous point because the performance of this step depends heavily on the implementation of the mechanism (e.g. a well-known address, a DNS-based approach, etc.) and possibly on the placement of the rendezvous point.

In the simulator, the locating processes of a joining node run in parallel and stop when one of them finds a group that meets the grouping criterion. A locating process also stops if its next hop would be a group it has already visited. If none of the locating processes are successful, the joining node gives up and creates a new group. Locating processes keep a list of visited groups. When a new group is created its neighbors are selected from the lists of all its locating processes. The first two joining nodes are special cases. They automatically create new groups because the grouping criterion cannot be evaluated without further nodes. As mentioned in Section 3, mOverlay does not define how to test if two distances are the same. However, we need to test this to check the grouping criterion. We have used test (1) from Section 3 also for the mOverlay simulation because it is a natural choice.

4.2 Meridian

In contrast to the mOverlay simulator, where we implemented all necessary messages, we did not implement the Meridian approach ourselves. Instead, we have used the official Meridian C++ implementation [3]. We have written wrapper code to redirect any messages to a simulation back-end instead of the network, and we have changed Meridian's time-keeping code to use the simulation time instead of the system clock. Each Meridian node is now a C++ object in the simulator rather than a physical node on the network. When it sends a packet the simulator determines the appropriate transmission latency using the underlying network model and schedules the packet arrival at the destination node accordingly. The wrapper objects also evaluate the grouping criterion at the end of a joining procedure and create a new group if necessary.

The simulation back-end is event-based. There are three kinds of events: one for inserting a new node into the scenario, one for triggering Meridian's periodic gossip protocol, and one for packet arrivals at a Meridian node. We start the simulation by scheduling node join events every seven seconds (which corresponds to Meridian's default gossip interval). When a node joins it starts by sending a closest node query to a Meridian node. This search is handled entirely by the original code. When the query returns, the joining node contacts the identified closest node to retrieve a list of verification nodes, which the wrapper code extracts from the Meridian object's latency cache. In the simulator we use a maximum of five verification nodes.

5 Simulation Results

5.1 Simulation Scenario

For the simulations we have used a matrix of round-trip times between 77 PlanetLab nodes, based on all-sites ping data from PlanetLab [5]. The simulator

estimates the one-way delay between two endpoints by dividing the appropriate round-trip time by two. At the time of writing, 694 machines hosted by 335 sites were part of PlanetLab. This means that each site hosts only slightly more than two machines on average. Consequently, we can expect to find groups of only a few nodes each in our scenario, especially since the 77 nodes in the network model were randomly selected from the available PlanetLab nodes. For each node pair we have also acquired a time series of round-trip times, which we use for evaluation. The time series contain round-trip time measurements made every 15 minutes during one day. We have originally planned to use more than 77 nodes, but we have removed several nodes from the original set due to missing values in the time series. Simulations have been run with different values for various parameters. Furthermore, each set of parameters was simulated using 100 different seeds, which we obtained from random.org [6].

5.2 Evaluation Criteria

We get the joining delays for every node, and the identified groups from a simulator run. While the comparison of the joining delays is straightforward, quantifying the quality of the identified groups is not. Grouping can exhibit two kinds of errors, false positives and false negatives. A node joining a group when it should not is considered a false positive and increases the error of grouping. A false negative occurs when a node erroneously does not join a group and creates a new one instead. This results in too many groups and impairs the efficiency and scalability of the overlay network. Unfortunately, due to the black box nature of our network model, we cannot say a priori whether a node should join a group or not. Nevertheless, we can define three criteria for the quality of the identified groups.

- First, the members of a group should be close to each other. Accordingly, we compute the mean round-trip time between members of the same group. Groups with only one node are ignored in this case.
- Second, bigger groups are preferable because they reduce the complexity of the overlay network. We use the average number of nodes per group as the second criterion.
- The third criterion stems from the use of the identified groups as a basis for a distance estimation service. One important assumption in mOverlay is that if two nodes A and B are in the same group, the distances $\overline{AC}$ and $\overline{BC}$ to a node C outside the group are virtually the same. This property enables significantly better scalability of the service. However, it must also hold over time. Otherwise, we would have to reorganize the groups constantly. We define the third criterion accordingly: If A and B are in the same group, $\overline{AC}_t$ should be a good prediction for $\overline{BC}_t$, where t is the time of measurement. We verify this using the time series of round-trip times between the two nodes. Two measurements $\overline{AC}_t$ and $\overline{BC}_t$ are out-of-band of each other if

$$\begin{array}{rlll} \overline{AC}_t \geq \overline{BC}_t & \wedge & (1-b)\cdot\overline{AC}_t > \overline{BC}_t \\ \vee\, \overline{AC}_t < \overline{BC}_t & \wedge & (1-b)\cdot\overline{BC}_t > \overline{AC}_t \end{array} \tag{2}$$

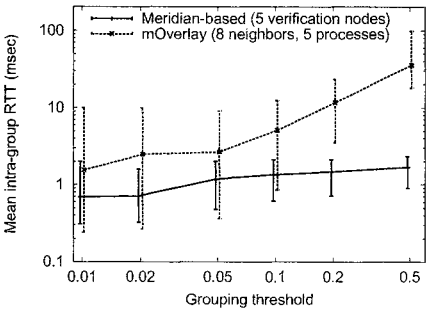


Figure 4. Mean intra group distance for several grouping thresholds

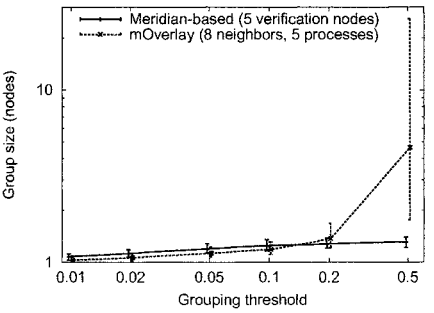


Figure 5. Avg. nodes per identified group for several grouping thresholds

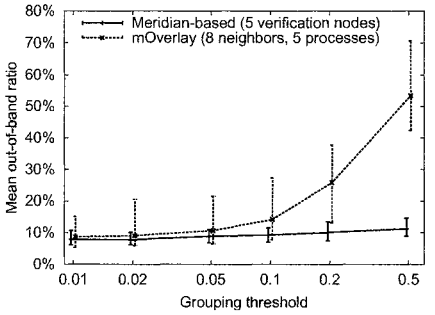


Figure 6. Mean out-of-band ratio using a 10% band, for several grouping thresholds

for a relative bandwidth $b \in [0, 1)$. The *out-of-band ratio* between two nodes is the ratio of out-of-band measurements in the respective time series. In this paper we use a bandwidth of 10% ($b = 0.1$).

The graphs in the remainder of this section use a dot-and-whisker format showing the mean with a 95% confidence interval, obtained by running the simulation with 100 different seeds. We have also slightly staggered the graphs along the horizontal axis to improve readability.

5.3 Quality of the groups

As a first comparison we look at the quality of the groups identified by mOverlay and our alternative approach. For both we use parameters that we have found to produce near optimal results. We set the maximum number of neighbors for mOverlay groups to eight and the number of parallel locating processes to five. For the Meridian-based approach we set the maximum number of verification nodes to five.

Figure 4 shows the mean round-trip times between group members for the grouping thresholds 1%, 2%, 5%, 10%, 20%, and 50% (using a logarithmic scale

for better readability). For both approaches a lower threshold also leads to smaller distances between group members. The effect is much bigger for mOverlay because the grouping threshold affects every iteration of the locating process, while the Meridian-based locating algorithm only uses the grouping threshold for its final step. Nevertheless, the round-trip times between group members of mOverlay are always bigger on the average than those of the Meridian-based approach. Moreover, the confidence intervals for mOverlay are bigger. We conclude that the Meridian-based approach performs better than mOverlay with respect to the first criterion.

The second aspect we examine is the average number of nodes per group. Figure 5 shows the group size for the same grouping thresholds as Figure 4. The groups identified by the alternate approach are bigger for grouping thresholds up to 10%. In contrast, mOverlay identifies much bigger groups with grouping thresholds above 10%, but this comes at the price of much greater round-trip times between group members. As expected, group sizes are rather small because of the wide distribution of the nodes.

If the identified groups shall be used as a basis for a distance estimation service they must also have a low out-of-band ratio. We look at this aspect using again the same parameters for grouping threshold and a 10% band for the out-of-band test. The results can be seen in Figure 6. The Meridian-based approach has shows a smaller out-of-band ratio than mOverlay for all grouping thresholds, and it shows less variance. Again, mOverlay shows high sensitivity towards the grouping threshold while the out-of-band ratio of the Meridian-based approach only increases slightly with growing grouping threshold.

5.4 Joining delay

In addition to a good quality of the identified groups it is also desirable to find the groups in the shortest time possible. We compare the two approaches using the same parameters as in Section 5.3. Figure 7 shows the joining delay per node for several grouping thresholds. Again, mOverlay proves to be much more sensitive towards the grouping threshold than the Meridian-based approach. Moreover, unless the grouping threshold is extremely high the alternate algorithm finds the local group much faster than mOverlay.

The joining delay of mOverlay nodes is not only sensitive to the choice of grouping threshold. Figure 8 shows the influence of the maximum number of neighbors per group and the number of parallel locating processes. Here we used a grouping threshold of 5% and a maximum of 2–10 neighbors per group. Furthermore, the three graphs show the effect of using 1, 5, or 10 parallel locating processes. We observe that a lower maximum of neighbors per group and fewer locating processes running in parallel cause a significant reduction in joining delay. Furthermore, the increase in joining delay appears to become smaller the more parallel locating processes we employ. However, regardless of the parameters the variance of the results is always quite large.

It appears that mOverlay can match the speed of the alternate approach if we reduce the number of parallel locating processes and the maximum number of

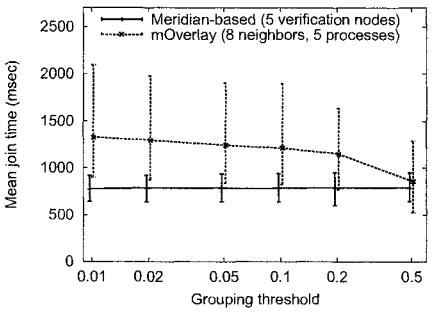


Figure 7. Mean joining delay per node for several grouping thresholds

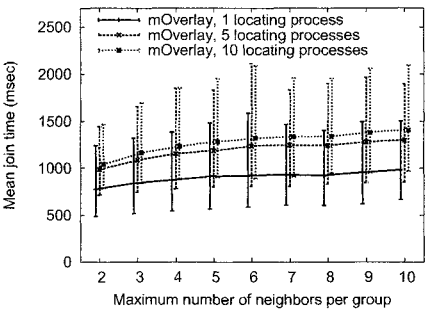


Figure 8. Mean joining delay in mOverlay for various parameters

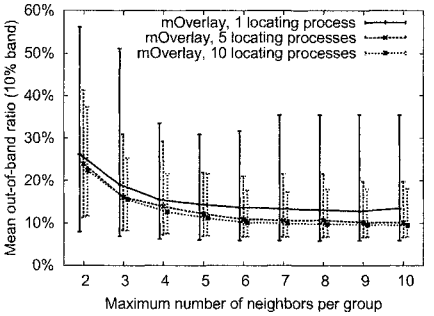


Figure 9. Mean out-of-band ratio in mOverlay for various parameters

neighbors per group. Nevertheless, the effect on the quality of the groups is severe as Figure 9 shows. Less than ca. 6 neighbors per group cause a significant increase in the out-of-band ratio. Using only one locating process also has a noticeable negative effect. On the other hand, the benefit from using more locating processes rapidly declines. The difference between five and ten parallel locating processes is mainly the size of the respective confidence intervals. Figure 9 also justifies our choice of parameters for mOverlay. The improvement for more than a maximum of eight neighbors per group and five locating processes is small while the increase in joining delay is still noticeable.

6 Conclusions

The locating algorithm of mOverlay has a number of drawbacks. We have presented an alternate algorithm that combines Meridian's closest node search with mOverlay's grouping criterion. In order to compare the mOverlay algorithm to the Meridian-based approach we have implemented simulators for each, using a black box network model based on PlanetLab measurements. We compare the

performance of both approaches based on the time it takes for a new node to find a group, the round-trip time between group members, the average number of nodes per group, and the suitability of the grouping for distance estimation, measured by the so-called out-of-band ratio.

From the simulation results we conclude that the Meridian-based locating algorithm is faster in most cases. It also identifies larger groups, and the nodes inside the groups are closer together than the nodes in mOverlay groups. Moreover, the groups identified with the alternate algorithm also have a smaller out-of-band ratio, which indicates better suitability for a distance estimation service.

References

1. X. Y. Zhang et al., "A construction of locality-aware overlay network: mOverlay and its performance," *IEEE JSAC*, vol. 22, no. 1, pp. 18–28, January 2004.
2. B. Wong, A. Slivkins, and E. G. Sirer, "Meridian: A lightweight network location service without virtual coordinates," ACM SIGCOMM'05, Philadelphia, Pennsylvania, USA, August 21–26, 2005.
3. Meridian C++ code, http://www.cs.cornell.edu/People/egs/meridian/code.php
4. M. Scheidegger, T. Braun, and F. Baumgartner, "Endpoint Cluster Identification for End-to-End Distance Estimation," ICC'06, Istanbul, Turkey, June 11–15, 2006, ISBN 1-4244-0355-3.
5. J. Stribling, "All-pairs-pings for PlanetLab," http://pdos.csail.mit.edu/~strib/pl_app.
6. random.org – True Random Number Service, http://www.random.org.
7. The Annotated Gnutella Protocol Specification, http://rfc-gnutella.sourceforge.net/developer/stable/index.html.
8. Clarke et al., "Protecting Free Expression Online with Freenet," IEEE Internet Computing, February 2002, pp. 40–49
9. S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, "A scalable content-addressable network," SIGCOMM 2001, Aug 2001.
10. S. Ratnasamy, M. Handley, R. Karp and S. Shenker, "Topologically aware overlay construction and server selection," IEEE Infocom'02, New York, NY, June 2002.
11. A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," IFIP/ACM Middleware 2001, Heidelberg, Germany, pp. 329–350, November 2001 .
12. L. Garcés-Erice, E. W. Biersack, and P. A. Felber, "MULTI+: Building topology-aware overlay multicast trees," in QofIS'04, September 2004.
13. M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," IEEE JSAC, Vol. 20, No. 8, pp. 1489–1499, 2002.
14. P. Francis, S. Jamin, J. Cheng, Y. Jin, D. Raz, and Y. Shavitt, "IDMaps: A global internet host distance estimation service," IEEE/ACM ToN, Vol. 9, No. 5, pp. 525–540, October 2001.
15. W. Theilmann and K. Rothermel, "Dynamic distance maps of the Internet," IEEE Infocom 2000.
16. T. S. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," ACM IMC 2003.
17. M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti, "Lighthouses for scalable distributed location," IPTPS 2003.
18. F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," ACM SIGCOMM 2004.

On Improving the Performance of Reliable Server Pooling Systems for Distance-Sensitive Distributed Applications

Thomas Dreibholz and Erwin P. Rathgeb

University of Duisburg-Essen, Ellernstrasse 29, 45326 Essen, Germany,
dreibh@iem.uni-due.de,
http://www.exp-math.uni-essen.de/~dreibh

Abstract. Reliable Server Pooling (RSerPool) is a protocol framework for server redundancy and session failover, currently under standardization by the IETF RSerPool WG. While the basic ideas of RSerPool are not new, their combination into a single, unified architecture is. Server pooling becomes increasingly important, because there is a growing amount of availability-critical applications. For a service to survive localized disasters, it is useful to place the servers of a pool at different locations. However, the current version of RSerPool does not incorporate the aspect of component distances in its server selection decisions.
In our paper, we present an approach to add distance-awareness to the RSerPool architecture, based on features of the SCTP transport protocol. This approach is examined and evaluated by simulations. But to also show its usefulness in real life, we furthermore validate our proposed extension by measurements in a PLANETLAB-based Internet scenario.

1 Introduction

The Reliable Server Pooling (RSerPool) architecture currently under standardization by the IETF RSerPool WG is an overlay network framework to provide server replication and session failover capabilities to its applications. These functionalities themselves are not new, but their combination into a single, unified and application-independent framework is.

While the initial motivation and main application of RSerPool is the telephone signalling transport over IP using the SS7 protocol [1], there has already been some research on the applicability and performance of RSerPool for other applications like VoIP with SIP [2,3], IP Flow Information Export (IPFIX) [4], SCTP-based mobility [5], real-time distributed computing [6–10] and battlefield networks [11]. But a detailed examination of an important application scenario is still missing: short transactions in widely distributed pools. Due to their short processing duration, network transport latency significantly contributes to their overall handling time. The goal of this paper is to optimize RSerPool's support for such transactions by extending RSerPool with an awareness for distances (i.e. latency) between clients and servers, as well as to define an appropriate server selection strategy trying to minimize this distance.

In section 2, we present the scope of RSerPool and related work, section 3 gives a short overview of the RSerPool architecture. A quantification of RSerPool

systems including the definition of performance metrics is given in section 4. This is followed by the description of our distance-sensitive server selection approach. Our approach is simulatively examined in section 6; experimental results in the PLANETLAB – showing the usefulness of our approach also in real life – are finally presented in section 7.

2 Scope and Related Work

A basic method to improve the availability of a service is server replication. Instead of having one server representing a single point of failure, servers are simply duplicated. In case of a server failure, a client's communication session can perform a failover to another server of the pool [7,12,13].

The existence of multiple servers for redundancy automatically leads to the issues of load distribution and load balancing. While load distribution [14] only refers to the assignment of work to a processing element, load balancing refines this definition by requiring the assignment to maintain a balance across the processing elements. This balance refers to an application-specific parameter like CPU load or memory usage. A classification of load distribution algorithms can be found in [15]; the two most important classes are non-adaptive and adaptive algorithms. Adaptive strategies base their assignment decisions on the current status of the processing elements (e.g. CPU load) and therefore require up-to-date information. On the other hand, non-adaptive algorithms do not require such status data. An analysis of adaptive load distribution algorithms can be found in [16]; performance evaluations for web server systems using different algorithms are presented in [17,18].

The scope of RSerPool [1] is to provide an open, application-independent and highly available framework for the management of server pools and the handling of a logical communication (session) between a client and a pool. Essentially, RSerPool constitutes a communications-oriented overlay network, where its session layer allows for session migration comparable to [19,20]. While server state replication is highly application-dependent and out of the scope of RSerPool, it provides mechanisms to support arbitrary schemes [7,12]. The pool management provides sophisticated server selection strategies [6,8,13,21] for load balancing, both adaptive and non-adaptive ones. Custom algorithms for new applications can be added easily [22].

3 The RSerPool Architecture

An illustration of the RSerPool architecture defined in [1] is shown in figure 1. It consists of three component classes: servers of a pool are called *pool elements* (PE). Each pool is identified by a unique *pool handle* (PH) in the handlespace, i.e. the set of all pools; the handlespace is managed by *pool registrars* (PR). PRs of an *operation scope* synchronize their view of the handlespace using the Endpoint haNdlespace Redundancy Protocol (ENRP [23]), transported via SCTP [24,25]. An operation scope has a limited range, e.g. a company or organization; RSerPool does not intend to scale to the whole Internet. Nevertheless, it is assumed that PEs can be distributed globally, for their service to survive localized disasters (e.g. earthquakes or floodings).

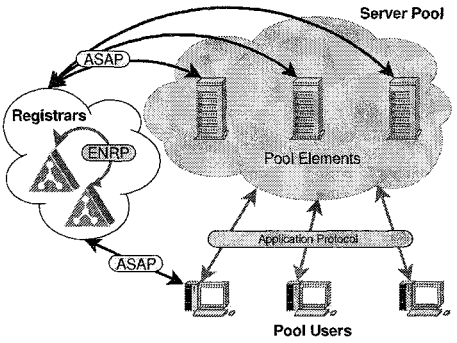


Fig. 1. The RSerPool Architecture

PEs choose an arbitrary PR to register into a pool using the Aggregate Server Access Protocol (ASAP [26]). Upon registration at a PR, the chosen PR becomes the Home-PR (PR-H) of the newly registered PE. A PR-H is responsible for monitoring its PEs' availability using ASAP Endpoint Keep-Alive messages (to be acknowledged by the PE within a given timeout) and propagates the information about its PEs to the other PRs of the operation scope via ENRP Update messages.

A client is called *pool user* (PU) in RSerPool terminology. To access the service of a pool given by its PH, a PE has to be selected. This selection – called *handle resolution* in RSerPool terminology – is performed by an arbitrary PR of the operation scope. A PU can request a handle resolution from a PR using the ASAP protocol. The PR selects PE identities using a pool-specific selection rule, called *pool policy*. A set of adaptive and non-adaptive pool policies is defined in [21]; for a detailed discussion of these policies, see [6, 8, 13, 22]. For this paper, only the adaptive Least Used (LU) policy is relevant. LU selects the least-used PE, according to up-to-date load information. The definition of *load* is application-specific and could e.g. be the current number of users, bandwidth or CPU load. For further information on RSerPool, see also [6–8, 13, 22, 27, 28].

4 Quantifying a RSerPool System

The service provider side of a RSerPool system consists of a pool of PEs, using a certain server selection policy. Each PE has a request handling *capacity*, which we define in the abstract unit of calculations per second. Depending on the application, an arbitrary view of capacity can be mapped to this definition, e.g. CPU cycles, bandwidth or memory usage. Each request consumes a certain amount of calculations, we call this amount *request size*. A PE can handle multiple requests simultaneously, in a processor sharing mode as commonly used in multitasking operating systems.

On the service user side, there is a set of PUs. The amount of PUs can be given by the ratio between PUs and PEs (PU:PE ratio), which defines the parallelism of the request handling. Each PU generates a new request in an interval denoted as *request interval*. The requests are queued and sequentially assigned to PEs.

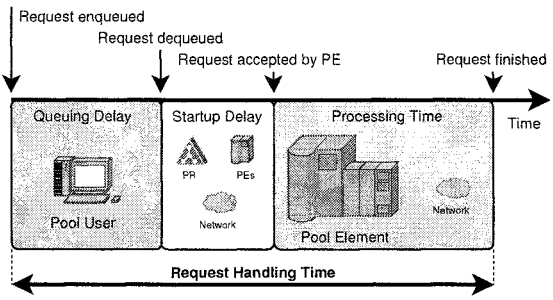


Fig. 2. Request Handling Delays

The total delay for handling a request d_{handling} is defined as the sum of queuing delay, startup delay (dequeuing until reception of acceptance acknowledgement) and processing time (acceptance until finish) as illustrated in figure 2. The *handling speed* (in calculations/s) is defined as:

$$\text{handlingSpeed} = \frac{\text{requestSize}}{d_{\text{handling}}}. \tag{1}$$

Clearly, the user-side performance metric is the handling speed – which should be as fast as possible.

Using the definitions above, it is now possible to give a formula for the system's utilization:

$$\text{systemUtilization} = \text{puToPERatio} * \frac{\frac{\text{requestSize}}{\text{requestInterval}}}{\text{peCapacity}} \tag{2}$$

Obviously, the provider-side performance metric is the system utilization, since only utilized servers gain revenue.

In summary, the workload of a RSerPool system is given by the three dimensions – PU:PE ratio, request interval and request size. In a well-designed client/server system, the amount and capacities of servers are provisioned for a certain *target system utilization*, e.g. 60%. That is, by setting any two of the parameters, the value of the third one can be calculated using equation 2. See [13] for a detailed discussion of the workload parameters.

5 A Distance-Aware Least Used Policy

As explained in section 1, PEs may be distributed over a large geographical area to survive localized disasters like an earthquake or tsunami. However, distributing PEs globally could e.g. result in PUs in Europe using PEs in Asia while PUs in America use PEs in Australia. Clearly, for transactions of short duration (compared to the network latency), this results in an increased overall request handling time. Currently, there are no distance-aware pool policies defined. Therefore, our goal is to adapt the Least Used policy to not only take care of PE load but also take the distance between PU and PE into consideration when selecting a server.

5.1 How to Quantify Distance?

Two approaches have been considered to actually quantify *distance*: using geographical position information and measuring the delay. Since geographically near endpoints do not necessarily have a low-delay connection (e.g. if using a satellite link), this approach is not useful. Instead, measuring the up-to-date network delay is preferable. Clearly, this implies the need for a measurement component. But in case of SCTP connections, this can be realized quite easily: the SCTP protocol [24] – used for the RSerPool communication – already calculates a smoothed round-trip time (RTT) for its paths. This RTT only has to be queried via the standard SCTP API [29]. Using the RTT, the end-to-end delay between two associated components is approximately $\frac{\mathrm{RTT}}{2}$.

In real networks, there may be negligible delay differences: for example, the delay between a PU and PE #1 is 5ms and the latency between the PU and PE #2 is 6ms. From the service user's perspective, such minor delay differences are negligible and furthermore unavoidable in Internet scenarios. Therefore, the distance parameter between two components A and B can be defined as follows:

$$\mathrm{Distance}_{A \leftrightarrow B} = \mathrm{DistanceStep} * \mathrm{round}\left(\frac{\frac{\mathrm{RTT}}{2}}{\mathrm{DistanceStep}}\right) \tag{3}$$

That is, the distance parameter is defined as the nearest integer multiple of the constant DistanceStep for the measured delay (i.e. $\frac{\mathrm{RTT}}{2}$).

5.2 An Environment for Distance-Aware Policies

In order to define a distance-aware policy, it is first necessary to define a basic rule: PEs and PUs choose "nearby" PRs. Since the operation scope of RSerPool is restricted to a single organization, this condition can be met easily by appropriately locating PRs. A PR-H can measure the delay of the ASAP associations to each of its PEs. As part of its ENRP updates to other PRs, it can report this measured delay together with the PE information. A non-PR-H receiving such an update simply adds the delay of the ENRP association with the PR-H to the PE's reported delay. Now, each PR can approximate the distance to every PE in the operation scope using equation 3. Note, that delay changes are propagated to all PRs upon PE re-registrations, i.e. the delay information (and the approximated distance) dynamically adapts to the state of the network.

5.3 The Policy Definition

As shown in [13], the Least Used policy provides the best performance and therefore becomes the obvious candidate to be extended with distance sensitivity: instead of only taking the load value into account for server selection, the new load value Load* is computed by increasing the PE's reported value Load by a distance-dependent *Distance Penalty Factor* (DPF) as follows:

$$\mathrm{Load}^* = \mathrm{Load} + \underbrace{\mathrm{Distance} * \mathrm{LoadDPF}}_{\text{Distance Penalty Factor}} . \tag{4}$$

The constant LoadDPF describes the load units per millisecond the actual load value is increased for every millisecond of the network delay. That is, the unit for LoadDPF is ms^{-1}. Due to the DPF parameter, the new policy is denoted as Least Used with DPF (LU-DPF). It simply selects the PE with the lowest value of Load*. If there are multiple lowest-valued PEs, round robin selection is applied among them.

Note, that the sorting of the PEs for selection is still per-PR rather than per-PU. This property is crucial for the efficiency of the handlespace management, since it allows for maintaining a LU-DPF pool by a set of PE identities sorted by Load* values. As shown in [22], this is very efficiently realizable.

6 Simulative Results

In order to examine the new policy, a simulative proof of concept has been performed first.

6.1 The Simulation Model

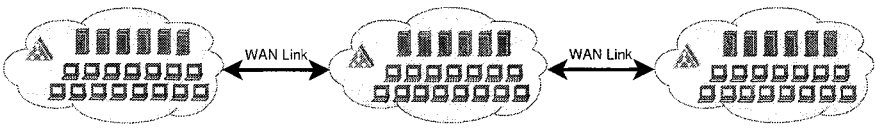


Fig. 3. The Simulation Setup

For our performance analysis, we have developed a simulation model using OMNeT++ [30], containing the protocols ASAP [26] and ENRP [23], a PR module and PE and PU modules modelling the request handling scenario defined in section 4. The simulation setup as shown in figure 3 consists of LANs interconnected by WAN links. Each LAN contains one PR and a variable amount of PEs and PUs – all in the same operation scope. As shown in [22], the component latencies are negligible and therefore have been omitted. As in [13], a negative exponential distribution is used for request intervals and sizes. For the policies, the *load* of a PE is defined as the current amount of simultaneously handled requests. The capacity of a PE is 10^6 calculations/s, the simulation runtime is 15 minutes; each simulation has been repeated 24 times with different seeds to achieve statistical accuracy. All results plots show the average values and their 95% confidence intervals.

6.2 A Proof of Concept

In our first simulation, we provide a proof of concept for the LU-DPF policy in a scenario consisting of 3 LANs, each containing 10 PEs. We have used a fixed inter-component LAN delay of 10ms and have varied the WAN delay from 0ms to 500ms (these settings are based on PlanetLab measurements and will be motivated in detail in subsection 7.1). The PU:PE amount ratio r varies from 1

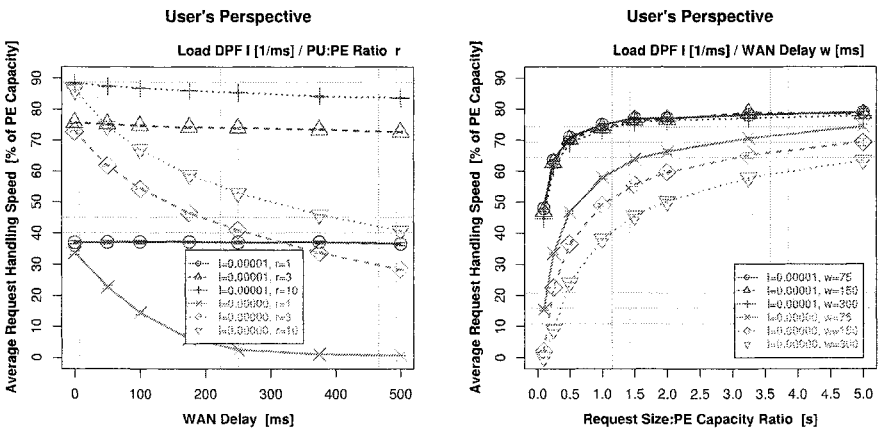


Fig. 4. A Proof of Concept

to 10 for DPF settings of 0 (i.e. the LU-DPF policy is equal to plain LU) and $1*10^{-5}$ (this parameter will be examined in detail in subsection 6.3); the average request size is 10^6 calculations (i.e. if processed exclusively, the processing time of an average request is 1s) and the target system utilization is 60% (the request interval is calculated using equation 2). The setting of DistanceStep is 1ms.

The left-hand side of figure 4 shows the resulting handling speed (in % of the PE capacity). As it is already shown in [6, 13], the PU:PE ratio r is the most critical load workload parameter. For smaller values of r, the per-PU load is highest, leading to higher performance degradation upon "bad" server selections. While the impact on the system utilization is negligible (therefore, the plot is omitted here), this effect is clearly visible for the handling speed – even if there is no WAN delay. As expected, the handling speed for a DPF of 0 (i.e. the policy behaves like plain LU) becomes smaller if the WAN delay increases. However, also taking the network delay into account (by setting the DPF to $1*10^{-5}$), the impact of the WAN delay becomes hardly visible. That is, our new LU-DPF policy has shown the desired effect: avoiding unnecessary delays by preferably using local PEs.

Obviously, the request handling speed for short transactions strongly depends on the network latency. To make this effect clearer, the right-hand side of figure 4 shows the speed results for varying the request size:PE capacity ratio for different WAN delay and DPF settings, using a PU:PE ratio of 3 and again a target system utilization of 60%. As expected, the handling speed for a DPF of 0 is significantly reduced by an increased WAN latency. However, if taking care of the delay by using a DPF of $1*10^{-5}$, the request handling speed becomes almost delay-independent. While it is obvious that the handling speed decreases with the request size for a DPF of 0, this effect can – in a smaller degree – also be observed for a DPF of $1*10^{-5}$: although the PUs preferably choose local PEs, there is still the inter-component LAN latency (10ms) which contributes to the startup delay of the requests. However, compared to the results of plain Least Used selection, the handling speed impact of small requests is significantly reduced (e.g. still a handling speed of 48% for a DPF of $1*10^{-5}$ vs. almost 0%

for a DPF of 0, at a WAN delay of 300ms for a request size:PE capacity ratio of 0.1). Therefore, the next question to be answered is: how to configure the DPF setting appropriately?

6.3 Configuring the Distance Penalty Factor

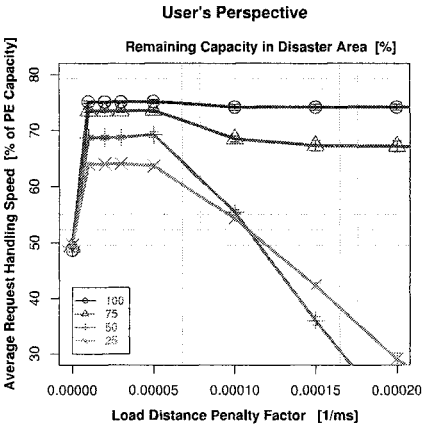


Fig. 5. Finding a Reasonable DPF Setting

After the first promising results from our proof-of-concept simulation, the following tasks have to be performed: (1) Find an appropriate DPF parameter setting and (2) verify that LU-DPF still provides a useful performance for the case that PEs in a region become unavailable (localized disaster) and remote PEs should be used instead.

To answer the questions, we have performed simulations for a large parameter space; the scenario presented here consists of a subset, carefully chosen to illustrate the essential effects. In this simulation, the DPF value is varied in a scenario consisting of 3 LANs with 12 PEs in each LAN, for a WAN delay of 150ms. Furthermore, the amount of PEs in the last LAN has been varied between 100% (i.e. all 12 PEs) and 25% (only 3 PEs) to simulate a localized disaster. To compensate the capacity loss in the last LAN, additional PEs have been equally distributed to the other 2 LANs. That is, the overall capacity of the pool always remains the same. All other parameters have been set as for the proof-of-concept simulation described in subsection 6.2.

While there is no significant impact on the utilization (therefore, we omit a figure), the handling speed as shown in figure 5 is significantly increased even for a small DPF setting. As expected, a higher value of the DPF setting has no impact if all PEs in the last LAN are available. For this scenario, the server selection mainly behaves as for three separate pools. However, decreasing the amount of PEs in the last LAN and increasing the amount in the other LANs, the scenario becomes heterogeneous. For setting the DPF parameter too high, the handling speed decreases and – for a sufficiently large setting (e.g. $15*10^{-5}$ for $\leq 50\%$ PEs in the last LAN) – the handling speed is even exceeded by plain LU (i.e. a DPF of 0).

In summary, the resulting general guideline on setting the DPF parameter is rather simple: set it to a value slightly above 0 – e.g. $1*10^{-5}$. In case of having multiple least-loaded PEs, this setting gives the server selection a preference for the nearest PE (see equation 4). Furthermore, it also provides an improved performance in scenarios of localized disasters – by enabling the selection of remote PEs if necessary. That is, LU-DPF can achieve a significant performance benefit over LU – at least in simulations. But since our goal is to also apply our new policy in real life, the next step is to validate our results by performing experiments in the Internet.

7 Experimental Results

Experimental results are necessary, because simulating all effects of the real Internet – including temporary QoS variations and the SCTP protocol's reaction – is almost impossible.

7.1 The Measurement Setup

In order to perform realistic measurements, we have used the PLANETLAB [31], a set of globally distributed hosts in the Internet. Based on our SCTP prototype implementation SCTPLIB [32] and our RSerPool implementation RSPLIB [27, 28, 33], we have realized an application model which is compatible to the simulated one. The setup consists of components distributed into three regions: Europe (mainly Germany), America (U.S.A., mainly West Coast) and Asia (mainly Japan). Each region contains one PR, which is used by the region's 5 PEs and 15 PUs. As for the simulations, the PEs have a capacity of 10^6 calculations/s; the PUs use a request size of 10^6 calculations and an average request interval of 7.5s (both using negative exponential distribution).

Tests using `ping` and `traceroute` have shown latencies between 5ms to 15ms within the regions; the inter-region delay varies between about 75ms to 150ms between Europe and America and America and Asia, as well as about 250ms to 350ms between Europe and Asia (routed via the U.S.A.). The delays between any two endpoints have not shown a significant variation. That is, it can be assumed that there has been sufficient bandwidth available. This is also realistic for RSerPool scenarios, since all components belong to a single operation scope (e.g. a company) and QoS mechanisms can therefore be applied easily (e.g. WAN connections via DiffServ-based VPN links using an appropriate SLA). Based on the delay experiments, DistanceStep has been set to 75ms.

7.2 Measurements

Each measurement run has a runtime of 65 minutes, with the following actions: at t_1=15min, 2 of the 5 Asian PEs are turned off; at t_2=30min, two additional PEs are turned on – one in America, the other one in Europe. At t_3=45min, the failure in Asia is repaired. Both PEs are again added to the pool, increasing its total capacity. The two additional PEs in Europe and America are turned off at

Interval	Network State	LU-DPF	LU	Improvement
1m - 14m	Normal Operation I	2.17s ± 0.05	2.63s ± 0.05	17.5%
16m - 29m	Failure in Asia	2.55s ± 0.02	2.78s ± 0.02	8.3%
31m - 45m	Added Backup Capacity	2.54s ± 0.02	2.71s ± 0.02	6.3%
46m - 49m	Failure Resolved	2.35s ± 0.06	2.55s ± 0.03	7.8%
51m - 64m	Normal Operation II	2.08s ± 0.02	2.47s ± 0.02	15.8%

Table 1. Average Request Handling Time Results

t_4=50min. In order to achieve sufficient statistical accuracy, the measurement has been performed 22 times, covering a total runtime of about 35 hours.

In order to compare the results for LU and LU-DPF (with a DPF setting of $1*10^{-5}$), we have performed both experiment runs simultaneously. That is, we have set up two pools – one for LU, the other one for LU-DPF. On each of the PE hosts, two PE instances have been started: one registering into the LU pool, the other one registering into the LU-DPF pool. Analogously, each PU host runs two PU instances: one using the LU pool, the other one using the LU-DPF pool. Due to the simultaneous execution, we ensure that both measurements are equally affected by temporal variations of the Internet's QoS conditions and keep the results comparable.

The resulting average request handling times and their 95% confidence intervals for both measurements are presented in table 1. Note, that we show the average over intervals beginning one minute after and ending one minute before a system condition change, since the latency to log into a PLANETLAB node to start or stop a component may take up to about 30s. Furthermore, small deviations of the hosts' clocks may be possible. Comparing the results for LU and LU-DPF, the LU-DPF policy provides a significant handling speed gain: between 17.6% and 15.8% for the two phases of normal operation and still around 8% during the failure and its resolution in Asia.

It is important to note that the performance in the "failure resolved" state is lower than for the "normal operation" states, although there are additional PEs in America and Europe: the over-capacity in these regions attracts the assignment of requests from Asia. This effect – the assignment of requests to slightly-loaded servers – is a property of all load-based policies; trying to avoid it by simply using a high LoadDPF setting would not lead to a performance improvement, as described in subsection 6.3.

In summary, the measurements have shown that our new LU-DPF policy is also working as intended in the real Internet.

8 Conclusion and Future Work

In this paper, we have presented a new, efficiently implementable, adaptive, distance-aware pool policy for RSerPool, which bases its server selection decisions on delay (measured by a feature of the SCTP protocol) and server load. The goal of this policy is to minimize the request handling time in situations where the processing time on the server is in the range of the network's transport latency.

To show the usefulness of our new policy, we have provided simulation results first. Furthermore, in order to also validate the policy's applicability in real life, we have performed measurements in the Internet using the PLANETLAB. Both

– simulations and measurements – have shown that our new policy can achieve a significant performance gain.

The future goal of our ongoing RSerPool research activities is to further examine the new policy under a broader range of network and application parameters – again, by simulations as well as measurements for validation. Furthermore, we intend to propose our new policy for standardization by the IETF RSerPool WG.

References

1. M. Tüxen, Q. Xie, R. Stewart, M. Shore, J. Loughney, and A. Silverton. Architecture for Reliable Server Pooling. Technical Report Version 11, IETF, RSerPool Working Group, March 2006. draft-ietf-rserpool-arch-11.txt, work in progress.
2. M. Bozinovski. *Fault-tolerant platforms for IP-based Session Control Systems.* PhD thesis, Aalborg University, Aalborg/Denmark, June 2004.
3. P. Conrad, A. Jungmaier, C. Ross, W.-C. Sim, and M. Tüxen. Reliable IP Telephony Applications with SIP using RSerPool. In *Proceedings of the State Coverage Initiatives 2002, Mobile/Wireless Computing and Communication Systems II*, volume X, Orlando, Florida/U.S.A., July 2002. ISBN 980-07-8150-1.
4. T. Dreibholz, L. Coene, and P. Conrad. Reliable Server pool use in IP flow information exchange. Internet-Draft Version 02, IETF, Individual Submission, February 2006. draft-coene-rserpool-applic-ipfix-02.txt, work in progress.
5. T. Dreibholz, A. Jungmaier, and M. Tüxen. A new Scheme for IP-based Internet Mobility. In *Proceedings of the 28th IEEE Local Computer Networks Conference*, pages 99–108, Königswinter/Germany, November 2003. ISBN 0-7695-2037-5.
6. T. Dreibholz and E. P. Rathgeb. The Performance of Reliable Server Pooling Systems in Different Server Capacity Scenarios. In *Proceedings of the IEEE TENCON '05*, Melbourne/Australia, November 2005. ISBN 0-7803-9312-0.
7. T. Dreibholz and E. P. Rathgeb. RSerPool – Providing Highly Available Services using Unreliable Servers. In *Proceedings of the 31st IEEE EuroMirco Conference on Software Engineering and Advanced Applications*, pages 396–403, Porto/Portugal, August 2005. ISBN 0-7695-2431-1.
8. T. Dreibholz, E. P. Rathgeb, and M. Tüxen. Load Distribution Performance of the Reliable Server Pooling Framework. In *Proceedings of the 4th IEEE International Conference on Networking*, volume 2, pages 564–574, Saint Gilles Les Bains/Reunion Island, April 2005. ISBN 3-540-25338-6.
9. T. Dreibholz and E. P. Rathgeb. An Application Demonstration of the Reliable Server Pooling Framework. In *Proceedings of the 24th IEEE INFOCOM*, Miami, Florida/U.S.A., March 2005. Demonstration and poster presentation.
10. T. Dreibholz. Applicability of Reliable Server Pooling for Real-Time Distributed Computing. Internet-Draft Version 01, IETF, Individual Submission, February 2006. draft-dreibholz-rserpool-applic-distcomp-01.txt, work in progress.
11. Ü. Uyar, J. Zheng, M. A. Fecko, S. Samtani, and P. Conrad. Evaluation of Architectures for Reliable Server Pooling in Wired and Wireless Environments. *IEEE JSAC Special Issue on Recent Advances in Service Overlay Networks*, 22(1):164–175, 2004.
12. T. Dreibholz. An Efficient Approach for State Sharing in Server Pools. In *Proceedings of the 27th IEEE Local Computer Networks Conference*, pages 348–352, Tampa, Florida/U.S.A., October 2002. ISBN 0-7695-1591-6.
13. T. Dreibholz and E. P. Rathgeb. On the Performance of Reliable Server Pooling Systems. In *Proceedings of the IEEE Conference on Local Computer Networks*

30th Anniversary, pages 200–208, Sydney/Australia, November 2005. ISBN 0-7695-2421-4.
14. E. Berger and J. C. Browne. Scalable Load Distribution and Load Balancing for Dynamic Parallel Programs. In *Proceedings of the International Workshop on Cluster-Based Computing 99*, Rhodes/Greece, June 1999.
15. D. Gupta and P. Bepari. Load Sharing in Distributed Systems. In *Proceedings of the National Workshop on Distributed Computing*, January 1999.
16. O. Kremien and J. Kramer. Methodical Analysis of Adaptive Load Sharing Algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 3(6), 1992.
17. M. Colajanni and P. S. Yu. A Performance Study of Robust Load Sharing Strategies for Distributed Heterogeneous Web Server Systems. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):398–414, 2002.
18. S. G. Dykes, K. A. Robbins, and C. L. Jeffery. An Empirical Evaluation of Client-Side Server Selection Algorithms. In *Proceedings of the IEEE Infocom 2000*, volume 3, pages 1361–1370, Tel Aviv/Israel, March 2000. ISBN 0-7803-5880-5.
19. F. Sultan, K. Srinivasan, D. Iyer, and L. Iftode. Migratory TCP: Highly available Internet services using connection migration. In *Proceedings of the ICDCS 2002*, pages 17–26, Vienna/Austria, July 2002.
20. L. Alvisi, T. C. Bressoud, A. El-Khashab, K. Marzullo, and D. Zagorodnov. Wrapping Server-Side TCP to Mask Connection Failures. In *Proceedings of the IEEE Infocom 2001*, volume 1, pages 329–337, Anchorage, Alaska/U.S.A., April 2001. ISBN 0-7803-7016-3.
21. M. Tüxen and T. Dreibholz. Reliable Server Pooling Policies. Internet-Draft Version 02, IETF, RSerPool Working Group, February 2006. draft-ietf-rserpool-policies-02.txt, work in progress.
22. T. Dreibholz and E. P. Rathgeb. Implementing the Reliable Server Pooling Framework. In *Proceedings of the 8th IEEE International Conference on Telecommunications*, volume 1, pages 21–28, Zagreb/Croatia, June 2005. ISBN 953-184-081-4.
23. Q. Xie, R. Stewart, M. Stillman, M. Tüxen, and A. Silverton. Endpoint Handlespace Redundancy Protocol (ENRP). Internet-Draft Version 13, IETF, RSerPool Working Group, February 2006. draft-ietf-rserpool-enrp-13.txt, work in progress.
24. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. Standards Track RFC 2960, IETF, October 2000.
25. A. Jungmaier. *Das Transportprotokoll SCTP*. PhD thesis, Universität Duisburg-Essen, Institut für Experimentelle Mathematik, August 2005.
26. R. Stewart, Q. Xie, M. Stillman, and M. Tüxen. Aggregate Server Access Protcol (ASAP). Technical Report Version 13, IETF, RSerPool Working Group, February 2006. draft-ietf-rserpool-asap-13.txt, work in progress.
27. T. Dreibholz. Das rsplib–Projekt – Hochverfügbarkeit mit Reliable Server Pooling. In *Proceedings of the LinuxTag*, Karlsruhe/Germany, June 2005.
28. T. Dreibholz and M. Tüxen. High Availability using Reliable Server Pooling. In *Proceedings of the Linux Conference Australia*, Perth/Australia, January 2003.
29. R. Stewart, Q. Xie, Y. Yarroll, J. Wood, K. Poon, and M. Tüxen. Sockets API Extensions for Stream Control Transmission Protocol (SCTP). Internet-Draft Version 12, IETF, Transport Area Working Group, February 2006. draft-ietf-tsvwg-sctpsocket-12.txt, work in progress.
30. A. Varga. OMNeT++ Discrete Event Simulation System, 2005.
31. Larry Peterson and Timothy Roscoe. The Design Principles of PlanetLab. *Operating Systems Review*, 40(1):11–16, January 2006.
32. M. Tüxen. The sctplib Prototype, 2001.
33. T. Dreibholz. Thomas Dreibholz's RSerPool Page, 2006.

Modeling Trust for Users and Agents in Ubiquitous Computing

Sebastian Ries*, Jussi Kangasharju, and Max Mühlhäuser

Department of Computer Science, Darmstadt University of Technology,
Hochschulstrasse 10, 64289 Darmstadt, Germany,
{ries, jussi, max}@tk.informatik.tu-darmstadt.de

Abstract Finding reliable partners for interactions is one of the challenges in ubiquitous computing and P2P systems. We believe, that this problem can be solved by assigning trust values to entities and allowing them to state opinions about the trustworthiness of others. In this paper, we introduce our vision of trust-aided computing, and we present a trust model, called CertainTrust, which can easily be interpreted and adjusted by users and software agents. A key feature of CertainTrust is that it is capable of expressing the certainty of a trust opinion depending on the context of use. We show how trust can be expressed using different representations (one for users and one for software agents) and present an automatic mapping to change between the representations.

1 Introduction

In [1], Bhargava et al. point out that "trust [...] is pervasive in social systems" and that "socially based paradigms will play a big role in pervasive-computing environments". Pervasive or ubiquitous computing is characterized by a very large number of smart devices, e.g., PDAs, mobiles, intelligent clothes etc., which come with different communication capabilities, storage, or battery power. Both the basic idea of ubiquitous computing and the heterogeneity of these devices call for interaction with and delegation to other devices.

Ubiquitous computing environments are unstructured and many service providers are available only locally or spontaneously. On the one hand, the interactions with foreign devices include uncertainty and risk, since a safe prediction of the behavior of those devices is not possible. On the other hand, the interactions with reliable partners are the basis for the services ubiquitous computing environments can provide. But how to select reliable interaction partners who behave as expected? Selecting only tamper-proof devices, which belong to the same manufacturer, requires the manufactures to be trusted, and unnecessarily reduces the potential of ubiquitous computing. Due to the great number of interactions with many different partners – some well-known, others not – and

* The author's work was supported by the German National Science Foundation (DFG) as part of the PhD program "Enabling Technologies for Electronic Commerce" at Darmstadt University of Technology

the claim of ubiquitous computing of being a calm technology, we need a non-intrusive way to cope with this challenge. We believe that the concept of trust, which has shown to work well in real life, is a promising solution, since it allows to make well-founded decisions in context of risk and uncertainty. Assuming recognition of entities, trust allows to express an expectation about the future behavior of an entity based on evidence from past engagements. Furthermore, trust needs representations which are meaningful not only to the software agents to enable automatic trust evaluation, but also to the end user, who needs to be able to understand the state of the trust model and to take part in the decision making process, if necessary.

In this paper, we provide a decentralized trust model, named *CertainTrust*, which allows agents to choose trustworthy partners for risky engagements. For our trust model, we propose two representations. The first one serves as a basis for a human trust interface. It represents trust using two independent parameters consisting of an estimate for the probability of trustworthy behavior in a future engagement, and of a parameter expressing the certainty of this estimate. Since we believe that trust is context-dependent, we also enforce the context-dependency of the certainty parameter of a trust value. The second representation is based on the Bayesian approach using beta probability density functions. This approach is well-established to express trust. It serves as a basis for the trust computation and as an interface for evidence-based feedback integration. Finally, we provide a mapping between both representations, and operators for 'consensus' and 'discounting'.

The remainder of this paper is structured as follows. In Sect. 2, we summarize our notion of trust and introduce our concept for the integration of trust in applications. Sect. 3 presents the trust model and the operators for trust propagation. In Sect. 4, we give an example showing how trust is represented and calculated using CertainTrust. Sect. 5 presents a summary of the related work, and Sect. 6 summarizes our contribution and outlines aspects of our future work.

2 Our Notion of Trust

From our point of view, trust is the well-founded willingness for a potentially risky engagement. Trust can be based on direct experience, recommendations, and the reputation assigned to the partner involved in an engagement. We model trust as the subjective probability that an entity behaves as expected based on experiences from past engagements.

2.1 Trust-Aided Computing

The integration of trust in ubiquitous computing applications seems to be a promising concept to cope with the new challenges of unstructured environments and dynamically changing interaction partners. In our vision of trust-aided computing (TAC) the applications are enabled to explicitly reason about trust. TAC

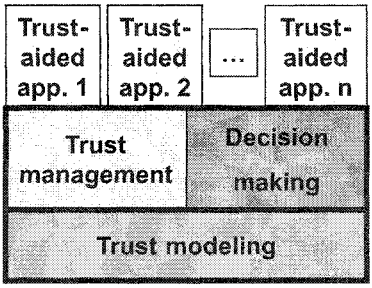


Figure 1. Trust-aided computing architecture

keeps track of the available entities, it collects information about direct experiences with other entities and information as recommendations and reputation. Thus, TAC allows applications to adapt to changes in the infrastructure and keep user preferences for interaction with entities which are already trusted. Not only relying on direct experience, but also on recommendations and reputation information, allows for building trust in entities with which no or only little direct experience is available. Since trust is subjective, this allows not only automated, but personalized decision making.

TAC disburdens the user from constantly being asked for the same decision in the same context. Furthermore, it disburdens the application developers from providing hard-coded strategies, which allow automated decision making, but cannot or can only hardly be personalized and do not take advantage of information collected in past engagements.

Trust-aided computing can be integrated in applications like P2P file sharing or packet routing in ad hoc networks to cope with malicious nodes. For ubiquitous computing trust-aided computing allows to find trustworthy devices in smart environments. On a higher level of abstraction trust-aided applications can help user to (semi-)automatically recognize trustworthy partners in virtual communities, e.g., to find reliable sellers in online auction platforms, or to evaluate recommendations provided by members of social networks (e.g., FilmTrust [4]). The trust-aided computing architecture has four main components (see Fig. 1):

- The *trust-aided applications* are able to reason about trust in a uniform way. They support the users since TAC allows to autonomously select trustworthy interaction partners. Yet, in critical cases, which can be identified, when reasoning about trust and risk, the user can be informed about the state of the system and asked for interaction.
- The *trust management component* focuses on the collection and filtering of evidence. Therefore, it is necessary to monitor the devices which are available, and to collect evidence from those devices. Furthermore, it is necessary to filter the collected evidences and recommendations based on policies to increase the level of attack-resistance. Other aspects of trust management are the evaluation of the context and the risk, which is associated to an engagement. Current trust management approaches are usually based on policies,

in which users can state which entities they consider as trustworthy. The approaches in [3, 5] already allow for the integration of trust levels.

- The *trust modeling component* concentrates on the reasoning about trustworthiness based on the available evidence. It provides the representational and computational models of trust. Since the users need to be able to set up, control and adjust the trust parameters, there is the need for a user interface, which can be used intuitively. Furthermore, there is the necessity for an interface, which is suitable for software agents allowing for automated integration of feedback and for autonomous evaluation of trust-relevant information. The computational model defines the aggregation of recommendations, reputation information and direct experience to a single opinion. For highly dynamic environments, which may contain a very large number of entities, e.g., ubiquitous computing environments, we cannot expect the end user to set up policies for each entity and for each context. Thus, the computational trust model itself has to provide some robustness towards attacks.
- The *decision making component* is often treated as a part of trust management. Since it is a very important aspect, and the users probably will judge the performance of a trust-based system, in the quality of its decisions, we like to mention it separately. The decision making has to consider the collected information about trust as well as the information about the expected risk. Although, we like to automate the decision process as far as possible, there must be the possibility of user interaction, to support the user to get used to TAC, and to be able to interact with the system in critical cases.

2.2 Properties of Trust

Having introduced an architecture which shows how to integrate trust in software applications, we focus on trust modeling for the rest of the paper. Our model expresses the following properties of trust: Trust is subjective, i.e., the trust of an agent A in an agent C does not need to be the same as trust of any other agent B in C. Furthermore, we cannot expect the behavior of A towards C to be the same as the behavior of C towards A, thus trust is asymmetric. Trust is context-dependent. Obviously, there is a difference in trusting in another agent as provider of mp3-files or as provider of an online banking service. It also is a difference in trusting in someone as service provider or as recommendation provider. If A trusts B in the context of providing recommendations about a good service provider, e.g., for file-storing, this does not necessarily imply that A trusts in B as a good peer to store files at, and vice versa. Trust is non-monotonic, i.e., experience can increase as well as decrease trust. Thus, we need to model both positive and negative evidence. Trust is not transitive in a mathematical sense, but the concept of recommendations is very important. Recommendations are necessary to introduce trust in agents with which no or only little direct experience is available. Moreover, we do not think of trust as finite resource, e.g., as done in flow-based approaches like EigenTrust [9]. It should be possible to increase trust in one entity without decreasing trust in another one.

2.3 Trust & Certainty

As in [6, 10, 12], we believe that it is necessary to express the (un-)certainty or reliability of an opinion stating trustworthiness. We also believe that the certainty of an opinion increases with the number of evidence, on which an opinion is based. Modeling the certainty of an opinion allows us to provide information on how much evidence an opinion is based, or to state that there is not any evidence available. Furthermore, it is possible to express that one opinion might be supported by more evidence than another one. We believe that the certainty value needs to be context-dependent because of the following reasons.

- In ubiquitous computing environments trust models can be used to automate decision making in many different contexts. In some contexts, there might be a great number of encounters, other contexts might be related to high risk, considering legal or financial implications. In these contexts, it seems reasonable, that users want to collect a great number of evidence, before they would think about an opinion to be certain. If forced to make a decision about an engagement involving high risk, one might choose to reject the engagement, although there is positive but too little evidence.
- In contexts in which the number of encounters is lower, or the associated risk is lower, users may be satisfied with a lower number of evidence to come to a well-founded decision.

To model the context-dependency for the certainty of an opinion, we assume there is a *maximal number of expected evidence* per context, which corresponds to the maximal level of certainty. For example, the maximal number of expected evidence can be defined as 5, 10, 100, or 1000.

3 Trust Model - CertainTrust

Let the contexts be denoted by con_i $i \in \{1, 2, \dots\}$, e.g., $con_1 = file_sharing$. For providing recommendations for a context con_i, we define a special context, which is denoted as rec_i. Let agents be denoted by capital letters $A, B, \dots$, and propositions by small letters $x, y, \dots$. The opinion of agent A about the truth of a proposition x in context con_i is denoted as $o_x^A(con_i)$. For example, in Fig. 2 the proposition x can be interpreted as $x =$ "Agent C behaves trustworthily in context con_i". The opinion of agent A about B's trustworthiness for providing recommendations for a context con_i is denoted as $o_B^A(rec_i)$. If the context is clear or not relevant, we use o_x^A and o_B^A. The maximal number of expected evidence (see Sect. 2.3) is denoted as $e(con_i)$ or e. Since the evidence model is partly derived from ideas presented in [6], we use the same terminology when possible.

We assume that evidence is collected and stored locally, and that recommendations are provided on request. The propagation of recommendations is done based on chains of recommendations, as shown in Fig. 2. We propose special operators for consensus (aggregation of opinions) and discounting (weighting of recommendations). For simplicity, opinions are assumed to be independent.

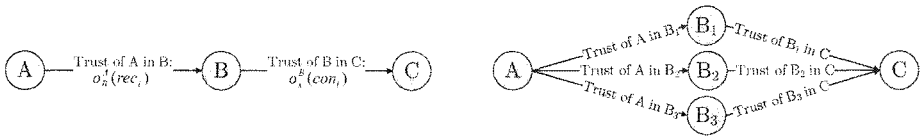


Figure 2. Trust chains

Our model provides two representations for opinions to express trust. The first representation is a pair of *trust value* and *certainty value* which serves as a base for a human trust interface. The second representation is based on the number of collected evidence and allows us to easily integrate feedback and forms the base of the computational model.

3.1 Human Trust Interface

The *human trust interface* (HTI) is used to represent trust as opinions. In the HTI an opinion o is a 2-tuple $o = (t, c)^{HTI} \in [0, 1] \times [0, 1]$, where HTI refers to the representational model. The opinion $o_B^A(rec_i) = (t_B^A(rec_i), c_B^A(rec_i))^{HTI}$ expresses the opinion of A about the trustworthiness of B in the context rec_i. The value of $t_B^A(rec_i)$ represents the probability that A considers the proposition "I believe, B to be trustworthy for providing recommendations for the context con_i" to be true. This value is the *trust value*. The value $c_B^A(rec_i)$ is the *certainty* or *certainty value*. This value expresses, which certainty the provider of an opinion assigns to the trust value. A low certainty value expresses that the trust value can easily change, and a high certainty expresses that the trust value is rather fixed. The values for trust and certainty can be assigned independently of each other. For example, an opinion $o_B^A(rec_i) = (1, 0.1)^{HTI}$ states that A expects B to be trustworthy in providing recommendations for con_i, but that A is not at all certain about this opinion.

For the moment, we express both the values for trust and certainty as continuous values in $[0, 1]$. Since humans are better in assigning discrete (verbal) values than continuous ones, as stated in [4,8], we want to point out, that both values can easily be mapped to a discrete set of values, e.g., to the natural numbers in $[0, 10]$, or to set of labels, as "very trusted" (vt), "trusted" (t), "undecided" (ud), "untrusted" (u), and "very untrusted" (vu). We assume that the trust and certainty values are independent, since the HTI then allows the users to easily express and interpret an opinion based on labels (see Fig. 3 (right side)). This is important since it allows users to check the current state of the model, and to set up and adjust those values according to personal preferences.

3.2 Evidence Model

The second representation, the evidence model, is based on beta probability density functions (pdf). The beta distribution $Beta(\alpha, \beta)$ can be used to model the posteriori probabilities of binary events. The beta pdf is defined by:

$$f(p \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \quad , \quad \text{where } 0 \leq p \leq 1,\ \alpha > 0,\ \beta > 0 \ . \tag{1}$$

Furthermore, we use $r = \alpha + 1$ and $s = \beta + 1$, where $r \geq 0$ and $s \geq 0$ represent the number of positive and negative evidence, respectively. The *number of collected evidence* is represented by $r + s$.

In the evidence model, an opinion o can be modeled using the parameters α and β. We denote this representation as $o = (\alpha, \beta)^{\alpha\beta}$. If the opinion is represented by the parameters r and s, we use the notation $o = (r, s)^{rs}$.

For $r + s \neq 0$ the mode t of the distribution $Beta(\alpha, \beta)$ is given as:

$$t = mode(\alpha, \beta) = \frac{\alpha - 1}{\alpha + \beta - 2} = \frac{r}{r+s} \tag{2}$$

For any $c \in \mathbb{R} \setminus \{0\}$ holds

$$mode((r, s)^{rs}) = mode((c \cdot r, c \cdot s)^{rs}) \ . \tag{3}$$

The main feature of this model is the easy integration of feedback in the trust model. Assuming that feedback fb can be expressed as real number in $[-1; 1]$, where '-1'is a negative experience and '1' is positive, the update of an opinion is done by recalculating the parameters $r_{new} = r_{old} + 0.5 * (1 + fb)$ and $s_{new} = s_{old} + 0.5 * (1 - fb)$ (cf. [7]). If the feedback is generated automatically, we can update the trust model without user interaction. Furthermore, the software agents can use all statistical information from the beta distribution, e.g., mean value and variance, as basis for decision making.

3.3 Mapping Between Both Representations

Trust value t of an opinion $o = (\alpha, \beta)^{\alpha\beta}$ is defined as the mode of the corresponding beta distribution. The certainty value c of an opinion $o = (\alpha, \beta)^{\alpha\beta}$ in a context con_i is defined as follows: The maximal number of expected evidence can be denoted by $e(con_i) = \alpha_{max} + \beta_{max} - 2$, where α_{max} and β_{max} fulfill:

$$mean_{coll} := \frac{\alpha}{\alpha + \beta} = \frac{\alpha_{max}}{\alpha_{max} + \beta_{max}} =: mean_{max} \tag{4}$$

Then the certainty c is calculated as:

$$c = \frac{f(mean_{coll} \mid \alpha, \beta) - 1}{f(mean_{max} \mid \alpha_{max}, \beta_{max}) - 1} \tag{5}$$

Definition 1 (Mapping) $(\alpha, \beta)^{\alpha\beta} = (t, c)^{HTI}$, *iff* $t = mode(\alpha, \beta)$ *and the certainty* c *fulfills Eq. 5.*

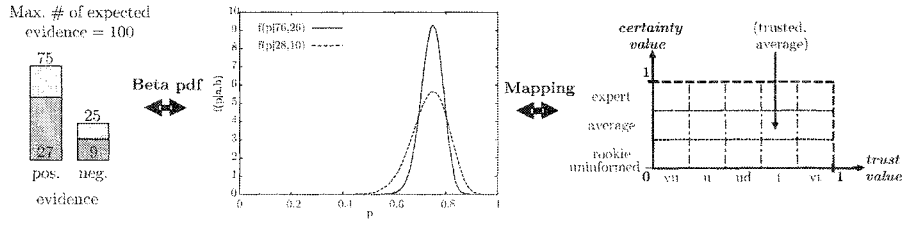


Figure 3. Mapping: Evidence - Agent Interface (pdf) - User Interface (HTI)

This mapping provides the translation between both representations (see Fig. 3). The interpretation of an opinion in the HTI by users has to be as close as possible to the interpretation of the same opinion in the evidence model by a software agent. This way, a user can interpret and adjust opinions based on evidence collected by a software agent and vice versa. Fig. 3 shows an opinion which is based on 27 positive and 9 negative pieces of evidence. The maximal number of expected evidence is 100. The mapping between the collected evidence and the agent interface is done using beta density functions. The mapping between the agent interface and the user interface is the one defined by Def. 1.

Intuitively, a human would set the trust value close to the observed relative frequency. Since the mode of a pdf is equal to the relative frequency of the observed event, the trust value t is close to the intuitive value set by the user. When no evidence is available, the mode of the pdf is not defined. In this case it can be reasonable to assume either a trust value of 0.5, or to infer the trust value based on the past experiences with formerly unknown entities.

The certainty value is intuitively linked to the number of collected evidence [6, 10,12]. A greater number of collected evidence leads to higher confidence, and to a higher certainty value. The maximal number of expected evidence $e(con_i)$ (see Sect. 2.3) is the maximal certainty value. Similar to [12], we want the certainty to increase adaptively with the number of collected evidence, i. e., the first pieces of evidence increase the certainty value more than later ones. As shown in Fig. 4, our certainty value fulfills these properties. In the absence of information ($r+s=0$), the certainty value is $c=0$, and $c=1$ if the number of collected evidence is equal to the expected number of evidence. Between the two extremes, the certainty value increases adaptively. If the number of collected evidence is greater than the number of expected evidence, there is a normalization, which preserves the trust value and scales the certainty to $c=1$ (see Eq. 6).

Normalization If an opinion $o=(r,s)^{rs}$ is based on more than the maximal number of expected evidence, it will be scaled to this maximum. The normalization preserves the mode of the pdf (see Eq. 3), and therefore, does not change the trust value. The normalized opinion $norm(o)$ will be used as input for the discounting described above.

$$norm((r,s)^{rs}) = \begin{cases} (r,s)^{rs} & \text{if } r+s \leq e \ , \\ (\frac{r}{r+s} \cdot e, \frac{s}{r+s} \cdot e)^{rs} & \text{else} \ . \end{cases} \tag{6}$$

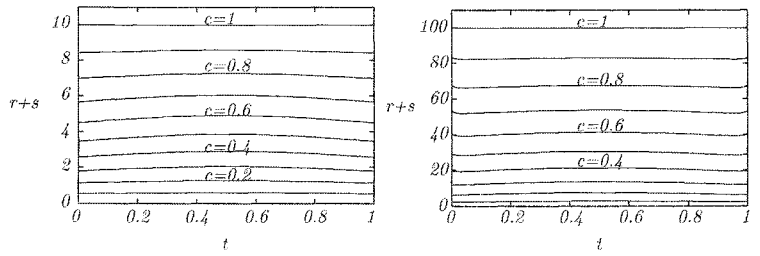


Figure 4. Iso-certainty lines: max. no. of exp. evidence $e = 10$ (l), $e = 100$ (r)

3.4 Trust Propagation

For trust propagation we define two operators, similar to the ones defined by Jøsang in [6]. We also call our operators 'consensus' for the aggregation of opinions and 'discounting' for the recommendation of an opinion. The consensus operator is identical with the one presented in [6]. We define a new discounting operator based on our evidence model.

Definition 2 (Consensus) *Let* $o_x^A = (r_x^A, s_x^A)^{rs}$ *and* $o_x^B = (r_x^B, s_x^B)^{rs}$ *be the opinions of* A *and* B *about truth of the proposition* x*. The opinion* $o_x^{A,B} = (r_x^{A,B}, s_x^{A,B})^{rs}$ *which combined the experiences of* A *and* B*, is defined as:*

$$o_x^{A,B} = o_x^A \oplus o_x^B = (r_x^A + r_x^B, s_x^A + s_x^B)^{rs} \tag{7}$$

The '$\oplus$' symbol denotes the consensus operator. The operator can easily be extended for the consensus between multiple opinions.

Definition 3 (Discounting) *Let* $o_B^A = (r_B^A, s_B^A)^{rs}$ *and* $o_x^B = (r_x^B, s_x^B)^{rs}$*. We denote the opinion of* A *about* x *based on the recommendation of* B *as* o_x^{AB} *and define it as:*

$$o_x^{AB} = o_B^A \otimes o_x^B = (d_B^A r_x^B, d_B^A s_x^B)^{rs} \quad , \text{where } d_B^A = t_B^A c_B^A \ . \tag{8}$$

The '$\otimes$' symbol denotes the discounting operator. In a chain of recommendations, we start at the end of the chain, e.g., $o_x^{ABC} = o_B^A \otimes (o_C^B \otimes o_x^C)$.

Discounting reduces the number of evidence taken into account, since $d_B^A \in [0; 1]$. The discounting factor d_B^A increases with positive evidence. That is, if A and C have the same amount of total evidence with B, but A has more positive evidence, then A gives a stronger weight to the recommendation of B than C.

Furthermore, the discounting factor increases with the number of collected evidence. That is, if A and C have the same ratio of positive and negative evidence with B, but A has more evidence in total, then A gives the opinion of B a stronger weight than C does.

Opinion	HTI	rs	Interpretation
A's opinions about B_i as recommenders			
$o^A_{B_1}$	(1,1)	(100,0)	(vt,expert)
$o^A_{B_2}$	(0.7,0.5)	(21.9,9.04)	(t,average)
$o^A_{B_3}$	(0.5,0.2)	(3.80,3.80)	(ud,rookie)
B_i's opinion about C as service provider			
$o_x^{B_1}$	(1,0.5)	(15.02,0)	(vt,average)
$o_x^{B_2}$	(0.7,0.5)	(11.19,4.80)	(t,average)
$o_x^{B_3}$	(0,0.5)	(0,15.02)	(vu, average)
Discounted opinions			
$o_x^{AB_1}$	(1,0.5)	(15.02,0)	(vt,average)
$o_x^{AB_2}$	(0.7,0.24)	(3.90,1.68)	(t,rookie)
$o_x^{AB_3}$	(0,0.10)	(0,1.50)	(vu,rookie)
Consensus: A's opinion about C as service provider			
$o_x^{AB_1,AB_2,AB_3}$	(0.86,0.62)	(18.92,3.18)	(vt,average)

Table 1. Example: Trust calculation

4 Example

Consider a file sharing scenario, where peers offer files for others to download. The risk in a file download is that the file might be corrupted or contain viruses. A simple corrupted file carries only a small risk, since we have only lost some CPU cycles and bandwidth, but a virus could potentially be extremely damaging. The trust is based on direct experience (i.e., downloads) and recommendations from other peers. In this example, we assume high risk and a high frequency of encounters. Therefore, we choose $e(con_i) = 50$ for the context of providing files, and $e(rec_i) = 100$ for the contexts of providing recommendations.

Peer A wants to download a file from peer C. Peer A has no direct experience with C, but A receives 3 recommendations about C's behavior (see right side of Fig. 2). Table 1 shows the collected and the calculated opinions.

This example shows how trust is represented and calculated in our model. Comparing the opinions $o_x^{B_i}$ and $o_x^{AB_i}$ shows that the discounting operation only manipulates the certainty values and preserves trust values. The consensus operation increases the certainty and adapts the trust value according to the given opinions. The resulting opinion $o_x^{AB_1,AB_2,AB_3}$ seems to be reasonable, when considering the input. Especially, we can see that the recommendation by B_3, which states C should be considered as "very untrustworthy", has only little impact on the resulting opinion, although all peers B_i claim to have a similar amount of experience with C. The reason is that the opinion $o^A_{B_3}$, stating that A has only little experience with B_3 with varying results, corresponds to lower discounting factor than the one of $o^A_{B_1}$ and $o^A_{B_2}$.

If A decides to download the file from C, then A can generate some feedback information based on the file's quality. This information can be used in an opinion as direct experience with C and to update the trust in the recommendations of the B_i's. If additional reputation information was available, it would have been integrated and discounted in the same way as the opinion of an agent B_i.

5 Related Work

The modeling of trust is addressed by a growing group of researchers [11]. There are several approaches which try to model trust one-dimensionally, e.g., Tidal-Trust [4] and EigenTrust [9]. A single trust value does not allow to express the certainty or the reliability of this trust value. Thus, it is impossible to express if an opinion is based on single evidence or on multiple pieces of evidence.

Other approaches model trust with two or three dimensions. Two dimensional trust models are often based on the Bayesian approach, e.g., [6, 10]. As stated in [8], those models are often too complicated to be understood by average users. The belief model approaches, e.g., [2, 6], use the triple belief b, disbelief d, and uncertainty u to represent trust. The problem with such models is that the three parameters cannot be assigned independently (e.g., $b + d + u = 1$ [6]) and hence the presence of uncertainty influences both belief and disbelief. Therefore, it is non-trivial to express, e.g., a medium belief with different levels of uncertainty. Our model allows to independently choose the values for trust and certainty.

Approaches like Subjective Logic [6] are not capable of expressing uncertainty context-dependently. In [6], the uncertainty u is defined as $u = 2/(r + s + 2)$. Therefore, uncertainty depends only on the number of collected evidence, but not on the context.

Other approaches presented in [10,12] introduce *reliability* as a concept which is similar to our concept of *certainty*. They also define a context-dependent value similar to the maximal number of expected evidence.

The model in [10] is based on the Bayesian approach. The maximal number of expected evidence e corresponds to m, which is described as the "minimal number of encounters necessary to achieve a given level of confidence". The reliability w is defined to increase linearly with the number of collected evidence from 0 (no evidence) to 1 (collected evidence at least m). But this linear approach is stated to be an first order approximation.

In the model in [12], the *intimate* level of interactions, is close to the concept of a maximal number of expected evidence. The *number of outcomes factor* (No) $\in [0, 1]$, increases with the number of collected evidence. To achieve the adaptive behavior as described in 3.3, the ratio between the collected and the expected number of evidence is used in a *sinus*-function.

6 Conclusion and Future Work

In this paper we introduced our vision of trust-aided computing, which allows for an explicit integration of trust information in applications. Furthermore, we provided a trust model, which allows to represent trust in a way, which can be interpreted and updated by software agents as well as by users.

We have shown, how to express the certainty of an opinion in contexts which are associated with different levels of risk, or frequency in interaction. In the HTI the values for trust and certainty can be interpreted independently, which allows to introduce the semantics of an opinion based on labels. Therefore, the

user is able to easily control the state of the trust model and to adjust opinions, if necessary. The evidence model enables software agents to update an opinion, when new evidence is available, and to reason about the trustworthiness of an interaction partner. The mapping between the HTI and the evidence model has an intuitive interpretation and it is mathematically founded. The two operators for trust propagation are based on the evidence model.

Our future work will include the development of trust management and decision making strategies. Those are necessary to be able to evaluate the trust model in a simulation, and to enhance the attack-resistance of the model. By first selecting recommendations from entities, which have repeatedly shown to provide good recommendations (high trust value and high certainty), and limiting the number of collected evidence to the maximal expected number of evidence, we believe to reduce the impact of sybil attacks to our trust model. Furthermore, we will refine the discrete representation for the human trust interface.

References

1. B. Bhargava, L. Lilien, A. Rosenthal, and M. Winslet. Pervasive Trust. *IEEE Intelligent Systems*, 19(5):74–88, 2004.
2. V. Cahill et al. Using Trust for Secure Collaboration in Uncertain Environments. *IEEE Pervasive Computing*, 2/3:52–61, July 2003.
3. M. Carbone, M. Nielsen, and V. Sassone. A Formal Model for Trust in Dynamic Networks. In *Proc. of IEEE Int. Conf. on Software Engineering and Formal Methods*, Brisbane, Australia, September 2003. IEEE Computer Society.
4. J. Golbeck. *Computing and Applying Trust in Web-Based Social Networks.* PhD thesis, University of Maryland, College Park, 2005.
5. T. Grandison. *Trust Management for Internet Applications.* PhD thesis, Imperial College London, 2003.
6. A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.
7. A. Jøsang and R. Ismail. The Beta Reputation System. In *Proceedings of the 15th Bled Conf. on Electronic Commerce*, June 2002.
8. A. Jøsang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. In *Decision Support Systems*, 2005.
9. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proc. of the 12th Int. Conf. on World Wide Web*, pages 640–651, New York, USA, 2003. ACM Press.
10. L. Mui et al. A Computational Model of Trust and Reputation for e-Businesses. In *Proceedings of the 35th HICSS*, 2002. IEEE Computer Society.
11. S. Ries, J. Kangasharju, and M. Mühlhäuser. A Classification of Trust Systems. In *On the Move to Meaningful Internet Systems 2006: OTM Workshops*, pages 894 – 903, Montpellier, France, 2006.
12. J. Sabater and C. Sierra. Reputation and Social Network Analysis in Multi-Agent Systems. In *Proceedings of the 1st Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 475–482, New York, NY, USA, 2002. ACM Press.

A Decentral Architecture for SIP-based Multimedia Networks

Holger Schmidt, Teodora Guenkova-Luy, and Franz J. Hauck

Institute of Distributed Systems, Ulm University, Germany
{holger.schmidt,teodora.guenkova-luy,franz.hauck}@uni-ulm.de

Abstract. The Session Initiation Protocol (SIP) is a general protocol for session setup and management, e.g., for VoIP. Current SIP networks build on a fixed infrastructure that relies on static network entities. General problems with this infrastructure, e.g., in ubiquitous ad-hoc networks, lead to a trend of integrating peer-to-peer mechanisms into SIP. We propose a generic architecture for decentralised SIP networks. The SIP location service of our architecture is based on JXTA for message routing. This allows SIP networks without central location services, and in extreme even without central SIP proxies. Unlike other approaches, we do not modify SIP. Thus, standard SIP clients can be seamlessly integrated. Due to the flexibility of JXTA, various peer-to-peer algorithms can be integrated according to current requirements. Additionally, our architecture supports registration, discovery and locating of services based on SIP. This saves resources in end terminals and also benefits from the peer-to-peer approach.

1 Introduction

There is a trend to integrate peer-to-peer (P2P) technologies into multimedia networks, especially into voice-over-IP (VoIP) networks, which traditionally rely on static network components. Consequently, the scalability is improved and single points of failure are avoided within the network architecture. One of the first and most popular realisation of P2P VoIP was implemented by Skype [1]. However, Skype builds on proprietary protocols, lacking interoperability with other VoIP platforms. VoIP providers push standard technologies as the Session Initiation Protocol (SIP) [2] to ensure the interoperability among different VoIP providers and systems.

SIP is a standardised protocol for session management (e.g., for establishing a VoIP call). For this task, it specifies central network entities, SIP proxies, which are responsible for SIP message routing (e.g., session establishment requests). As SIP seems to be the first choice for VoIP session management (e.g., SIP is the protocol for session management in the 3rd Generation Partnership Project—3GPP [3]), there are efforts to integrate P2P technologies with the SIP technology. There are two approaches in general. One possibility is the adoption of the protocol to P2P networks by adding overlay network information to the protocol headers [4]. Another approach is the integration of an existing

P2P technology into the SIP user-agent logic, resulting in no changes to the SIP headers specification. In the second case, the registration and the discovery of terminals is done using the overlay P2P network, thus making the application of centralised SIP proxies unnecessary. In this work, we target the latter approach.

Recently, we integrated service location into standard SIP [5], as service location is often a crucial part of session establishment [6, 7]. Consequently, advantages are achieved, especially in ubiquitous computing environments with mobile, resource-limited devices. Instead of two stacks—one for service location and one for session management—only one integrated SIP stack is needed. Furthermore, we are able to insert service location information into standard SIP messages which reduces network traffic by decreasing the number of messages (see Section 2.2).

In ubiquitous computing environments, in which spontaneous communication among the mobile distributed devices dominates, service location is an essential part. Whenever a mobile device enters a new access network or in case it wishes to communicate with other devices within a network, the device tries to discover available services in its neighbourhood, e.g., searching for multimedia proxies that are able to convert multimedia-data formats and configurations in case the terminal has no shared audio/video capabilities with its communication partners [6, 7]. Our initially developed concept for SIP service location relies on static designated network entities for service discovery [5]. In this paper, we develop this concept further and provide an approach that is built on standard technologies for a decentralised service location with SIP using P2P mechanisms. P2P-based SIP can be especially useful for ubiquitous ad-hoc networks [8]. For integrating P2P mechanisms into SIP, we build on the open source platform JXTA [9] that allows the flexible implementation of structured—distributed hash table (DHT)-based—and unstructured routing mechanisms. In contrast to former work, we think that supporting unstructured P2P networks is very important, as service discovery should be able to efficiently handle range queries to service groups as well (e.g., to manage non-exact queries and to evaluate alternative service possibilities), as most DHT-based systems can only handle previously indexed exact-match queries efficiently. Furthermore, service descriptions can have arbitrary formats and in this case cannot be indexed efficiently in dynamic environments. Additionally, nodes in DHT-based P2P systems are characterized by processing a higher maintenance traffic [10]. Especially on mobile devices, a configurability of the P2P routing mechanism can save resources. By building on a P2P infrastructure as a data store, better scalability is achieved and coping with node-crashes (fail-stop behaviour) can be improved.

The structure of this paper is as follows. After giving basic background information on SIP and our SIP service location mechanism in Section 2, we discuss related work in Section 3. Then, we present our JXTA-based architecture for decentralised SIP Service Location in Section 4. In Section 5, we provide performance measurements, and finally Section 6 concludes and shows possible future work.

2 Preliminaries

2.1 Session-Initiation-Protocol Fundamentals

SIP was developed by the Internet Engineering Task Force (IETF) [2]. The protocol is used for session management and coordination in general. Currently, SIP is used predominantly for session management of multimedia sessions, i.e., for establishing, modifying and terminating VoIP sessions.

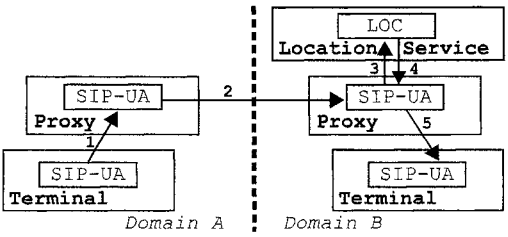


Fig. 1. Message flow for SIP session establishment

SIP is a text-based application-layer protocol that builds on the client/server communication paradigm. The SIP RFC 3261 [2] specifies the behaviour of several network entities: user agents, registrars, location services and proxies. A user agent (UA) represents a terminal that is able to establish, to modify, and to terminate sessions with other UAs. For supporting message delivery to other UAs, SIP proxies are used (cf. Figure 1). A SIP proxy is a network entity that controls the message flow by forwarding the message to the nearest known entity. If the target is not in the same domain, the message is just forwarded to a proxy that is responsible for the specific domain. The address of the responsible proxy is obtained by retrieving an SRV record from the Domain Name System (DNS) [11]. If the target is in the proxy's domain, the SIP location service (LOC) is used, that stores contact information about all previously registered UAs of a specific domain. Thus, the location service is able to return the final address of the UA, which is used by the proxy to forward the message.

2.2 SIP Service Location

Recent SIP entities rely on using special protocols for service location, e.g., the Service Location Protocol (SLP) [12]. However, as in most cases multimedia communication coordination is already based on SIP, an integrated service location into the SIP protocol saves resources as we outlined in [5]. As described in Section 2.1, terminals have to register their contact information first. As SIP messages can carry so-called bodies with arbitrary content, the SIP registration message (*REGISTER*) can be used to additionally transfer information about

```
REGISTER sips:registrar.uulm.de SIP/2.0
Via: SIP/2.0/TLS client.uulm.de:5061;
     branch=z9hG4bKnashds7
Max-Forwards: 50
From: Bob <sips:bob@uulm.de>;tag=a73kszlfl
To: Bob <sips:bob@uulm.de>
Call-ID: 1j9FpLxk3uxtm8tn@uulm.de
CSeq: 1 REGISTER
Contact: <sips:bob@client.uulm.de>
Content-Type: text/directory;
              profile="x-slp"
Content-Length: 116

URL: service:lpr://www.uulm.de:598/xyz
Attributes: (SCOPE = STUDENTS),
            (PAPERCOLOR = YELLOW),
            (PAPERSIZE = A4)
```

Fig. 2. REGISTER-request containing SLP service description of a printer service

locally available services together with the device's contact location information (Fig. 2 shows a registration carrying SLP content describing a printer service). In this case, the registrar also acts as a node that stores service descriptions in the location service for later search requests. This enables an efficient registration using just one processing step for both types of registrations. In case of a separate service-location infrastructure, the UA would have to register its services after having registered its SIP identity at the registrar.

Service-location information is automatically kept up-to-date by subsequent *REGISTER* requests, as UAs have to re-register periodically according to the SIP standard. This ensures a clean data basis in case of node crashes. The service-location information is automatically deleted if the UA deregisters.

We integrated service discovery into the SIP protocol as well. Therefore, a standard SIP *OPTIONS* message is used to transfer queries as an attachment. This message is sent to the service location server[1], which is able to interpret the SIP body, query the location service and, then, return appropriate service results in the attachment of a *200 OK* response message.

Using only SIP for service location and session management leads to less code at the UA's side, as there is no more need for a special service-location protocol implementation. However, some special functions for service registration and discovery have to be added on top of a standard SIP implementation. For typical ubiquitous computing scenarios with small, mobile, and resource-limited devices, service location is an essential step whenever entering a new environment [13]. There, our SIP-only solution is an advantage, as memory and computing power can be saved. Finally, our approach for SIP service location can be used for discovery of devices with specific capabilities and devices at specific locations as well [5].

[1] This entity can be integrated into the SIP registrar as well

3 Related Work

Currently, integrating P2P technologies into the SIP protocol is a popular area of research. Within the Internet Engineering Task Force (IETF), there are many discussions about starting a P2P SIP charter soon. In this group, concepts, terminology, and the infrastructure for SIP in a P2P environment are discussed [14]. Especially, remaining burdens are discussed, e.g., the allocation and the protection of SIP names in a distributed infrastructure and general requirements for P2P SIP [15].

In general, there are two approaches for integrating P2P technologies into SIP. These are described in [16]: Either P2P protocols are integrated into the SIP protocol or the SIP location service builds on a P2P infrastructure.

Bryan et al. [4] presented a P2P-based approach for SIP registration and resource discovery that is based on Chord [17]. This concept removes the need for central SIP entities and provides backward compatibility. However, in contrast to our work, there is a need for a special P2P-SIP protocol implementation to participate in the P2P overlay. Furthermore, the draft specifies a hard-wired support for Chord, other P2P protocols are not supported. Our infrastructure builds on the standardised and open platform JXTA, that allows the seamless integration of arbitrary routing mechanisms. This is especially important for ubiquitous mobile devices, as the routing mechanism affects resource usage by a higher maintenance traffic [10]. Additionally, if service location is integrated, unstructured P2P overlays provide a more efficient support for range queries.

Johnston et al. [18] discussed the application of P2P technologies to SIP. They propose a standardisation of a location service protocol to address the location service from SIP proxies/registrars. They outline that this protocol does not need to be necessarily a P2P protocol. We do not specify a special location service protocol; in contrast we use basic JXTA protocols for storing and loading data into the JXTA network.

Singh and Schulzrinne [19] developed a P2P-based outbound-proxy that intercepts standard SIP messages and provides a P2P infrastructure for registration and discovery. They build on a modular design and use Chord as P2P protocol. However, our approach provides a more generic architecture for the seamless integration of P2P technology into SIP by building on standard SIP and the standard JXTA architecture, in which any routing mechanism can be integrated as well (important for mobile devices and when integrating additional services).

In contrast to all the above-mentioned state-of-the-art approaches, we integrated a P2P-based solution for service location in a SIP network. Especially in ad-hoc P2P networks, where service location is an essential step, our technique results in better scalability and fault tolerance among the participating entities.

4 Architecture for Decentralised SIP Service Location

In this section, we introduce the JXTA P2P platform and then present a JXTA-based generic architecture for integrating P2P technology into a SIP network.

Last, we describe the integration of our developed SIP service location infrastructure into this architecture.

4.1 JXTA Fundamentals

The JXTA project is a generic and open source platform for P2P applications [9]. As most of these P2P applications share fundamental concepts, JXTA tries to establish a common basis for future interoperable applications. Therefore, JXTA specifies six protocols that build on asynchronous query/response communication [20].

The JXTA network is organised into peer groups, which consist of several peers representing nodes within the JXTA network. Each peer belongs to at least one peer group, which restricts the propagation of communication traffic to group members. Like any resource in the JXTA network, these peers have unique identifiers (IDs) and are actually represented by an XML metadata structure called *advertisement*.

There are different types of peers, e.g., standard nodes are called *edge peers* and super nodes are called *rendezvous peers*. A peer searches for network resources by seeking for an appropriate advertisement. These advertisements are published for a specific lifetime within a certain peer group [21]. For supporting the discovery process, peers publish advertisements at rendezvous peers, which create an index of the edge peer's resources. This allows efficient search for advertisements, as a peer first tries to discover advertisements using broadcast within the local network and in case of no success the advertisement is searched via the index data of the rendezvous peer network. However, JXTA allows the integration of arbitrary custom routing mechanisms (e.g., DHT-based).

4.2 Decentralised SIP Network Entities using JXTA

In contrast to former work [4], we do not integrate P2P techniques directly into the SIP protocol as standard SIP-capable devices could not participate in such a P2P overlay network. For a seamless integration, we developed a SIP location service that stores data in a P2P overlay network by using JXTA. JXTA enables the integration of arbitrary P2P routing mechanisms, i.e., structured and unstructured routing, which allows an adaptation to current environment's needs (e.g., service location needs range queries—unstructured P2P overlays are more efficient in this case; see Section 4.3). This location service implementation is used by the registrar to store registration data inside the P2P network and by the SIP proxies to obtain routing information. Thus, the registrar and the SIP proxy act like standard entities that internally use our special location service. The proxy and the registrar are installed and addressed as local entities.

As standard SIP location services are responsible for a certain domain, we transferred this concept to our JXTA location service (JXTA-LOC). Our location service connects itself to a predefined peer group in which information about UA registrations are published and discovered. Thus, we create a kind of virtual overlay for SIP domains, which are projected onto JXTA peer groups (The

name of the peer group is the name of the domain with the prefix 'sip:'). We specified a custom advertisement (cf. Fig. 3) representing information about registered UAs. This advertisement is published whenever a SIP user registers and it contains a unique identifier (`Id`), the registered SIP address (`Key`), the advertisement type (`Type`), and actual contact addresses (`Contacts`).

```
<!DOCTYPE jxta:SipRegAdvertisement>
<jxta:SipRegAdvertisement xmlns:jxta="http://jxta.org">
  <Id>urn:jxta:uuid-0146...5DD301</Id>
  <Key>sip:sender@uni-ulm.de</Key>
  <Type>jxta:SipRegAdvertisement</Type>
  <Contacts>
    <Contact>sip:sender@134.60.77.162:5060</Contact>
  </Contacts>
</jxta:SipRegAdvertisement>
```

Fig. 3. Advertisement for registration information

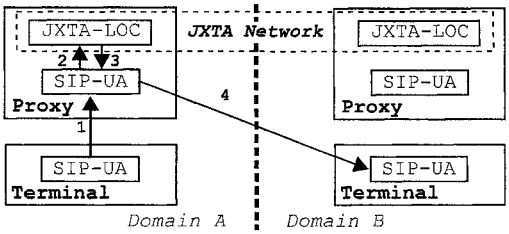


Fig. 4. Scenario 1: Session establishment via proxy with integrated JXTA location service (classic terminals)

SIP registrations have a certain expire time. Thus, these registrations should be available in the P2P overlay for this time only. JXTA provides an expire time for advertisements which is set according to the SIP *REGISTER* request. By using the advertisement's unique identifier, it can be updated and deleted.

Whenever a SIP proxy gets a request message, it extracts the receiver's SIP address, and then queries the JXTA network in the appropriate peer group (receiver's domain with 'sip:' prefix). If the receiver's contact address is found, the proxy is able to forward the message directly to the receiver (cf. Fig. 4). If the SIP proxy is not responsible for this domain, it forwards the request to the domain's responsible SIP proxy (see Section 2.1). When the proxy gets a timeout while trying to forward the message, it queries the JXTA network for an updated registration (advertisement). If an updated advertisement is found, the proxy tries to forward the message once more, otherwise, an error response is sent to the initiating UA.

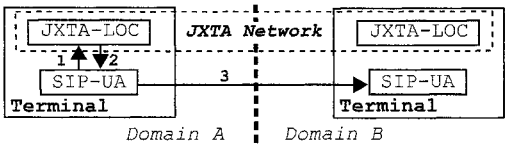


Fig. 5. Scenario 2: Session establishment of a user agent with integrated JXTA location service (JXTA terminals)

The JXTA location service can be integrated into the UA as well (cf. Fig. 5). This results in no further need of proxies and registrars within the network. For registration of the local contact address, the UA uses the JXTA location service for publishing an appropriate advertisement. For establishing a session, the potential receiver's contact address is determined by the JXTA infrastructure. Then, the session establishment request (*INVITE*-message) is sent directly to the actual receiver's address. Therefore, this kind of session establishment is even more efficient, as the forwarding of messages by SIP proxies is no longer necessary, but the UA has to be adapted.

4.3 Integration of Service Location

The decentralised service location infrastructure builds on our recently developed service location infrastructure and the decentralised JXTA-based SIP network. As in a standard SIP environment, UAs register their contact information at the registrar. In our service location infrastructure, UAs are able to attach information about their provided services. The registrar parses these data and stores them by publishing an extended custom advertisement, which exists for a certain limited time according to the SIP registration's expire time. (cf. Fig. 6). This advertisement comprises the same information as the advertisement presented in Section 4.2. Additionally, it contains arbitrary typed metadata about available services (`Contents`), e.g., a printer service that is available at the UA's host (cf. Fig. 6). As these metadata is not standardised, using a DHT-based P2P overlay is not efficient, as an overall key-generation algorithm for specifying the actual distribution cannot be defined. JXTA supports an unstructured P2P routing mechanism. If the metadata is specified in a closed environment, JXTA allows implementing a structured P2P overlay as well. However, unstructured P2P overlays are especially valuable for range queries, e.g., when searching for services within certain location coordinates based on GPS-data.

For searching for available services, a UA first has to find possible service location servers (SLS). Therefore, the registrar returns a *200 OK* message that contains one or more contact headers with a SIP- or a SIPS-URI with the parameter *servicelocation=true* and supported service description formats (e.g., *Contact: <sip:sls@uulm.de; servicelocation=true; serviceformat=slp;>*). These URIs identify the SLSs within the current domain.

```
<!DOCTYPE jxta:ExtSipRegAdvertisement>
<jxta:ExtSipRegAdvertisement xmlns:jxta="..." >
  <Id>... </Id>
  <Key>...</Key>
  <Type>jxta:ExtSipRegAdvertisement</Type>
  <Contacts>...</Contacts>
  <Contents>
    <Content type="text/directory;profile=x-slp">
      URL: service:lpr://www.uulm.de:598/xyz
      Attributes: (SCOPE = STUDENTS),
                  (PAPERCOLOR = YELLOW),
                  (PAPERSIZE = A4)
    </Content>
  </Contents>
</jxta:SipRegAdvertisement>
```

Fig. 6. Extended advertisement with registration information

A service discovery is initiated by the UA by sending an *OPTIONS* message containing the query to an SLS. The SLS parses this query and then searches for appropriate advertisements in the JXTA network. If advertisements are found, a *200 OK* response is returned that includes the results.

For keeping information of a UA up-to-date, service information has to be republished periodically. As standard SIP UAs have to periodically register their contact information, this message is used to attach service information as well. The contact and service information is stored in the JXTA network by re-publishing the advertisement. Re-publishing actually means searching for the advertisement containing the registered SIP URI, and then publishing an advertisement containing the found advertisement's `Id`.

Service information of a UA is deleted if it registers without attaching service information. If a UA deregisters, the advertisement is deleted from the network.

In case of an integrated location service within the UA, the UA is able to publish and to discover advertisements without the need of a central SIP entity.

5 Performance Measurements

In this section, we provide performance measurements that show the benefit of our approach of integrating P2P technologies into SIP networks using JXTA. Therefore, we implemented a proxy server using the NIST SIP [22] stack. Our proxy server is able to use our JXTA location service with the standard JXTA routing mechanism (Section 4.1). Alternatively, an integrated location service implemented in Java `Hashtable` for storing SIP contact information is used.

We compared the scenario of Fig. 1 (Standard) with the scenario of Fig. 4 (P2P). These scenarios are comparable to a company, which operates at different locations. In this case, internal sessions can be carried out by the P2P overlay network. The SIP network entities of domain A and of domain B, each reside on a machine with an AMD Athlon XP 2100+ processor and 1GB RAM. A switched 100MBit ethernet network connects both machines.

We implemented UAs for sending and for receiving *MESSAGE* requests for simple text mesages. First, these UAs register at their domain-specific registrar,

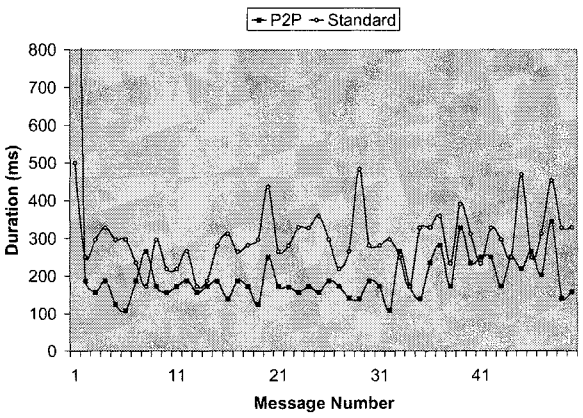


Fig. 7. Time consumed for successful messages from a sender to a receiver using P2P vs. standard SIP location service

then one UA acts as a sender for *MESSAGE* requests, and the other UA acts as a receiver. In the first test series, the registrar uses the JXTA-based location service. Then, the Java `Hashtable`-based location service is used. We measured in each case the time consumed from sending a message until receiving the response. Fig. 7 shows the results.

The first request within the P2P infrastructure takes about four times longer than the request in the Standard-SIP infrastructure. This can be explained by the JXTA discovery process. The JXTA location service implementation has to search for locally not existent advertisements within the JXTA infrastructure. This takes some time, however, these advertisements are subsequently cached for future use. This results in shorter discovery times at later searches (cf. Fig. 7).

The P2P infrastructure reduces inter-proxy communication, as the UA's actual address can be retrieved within the JXTA network. Thus, the total time consumed for a successful message request and its response is lower than in the standard SIP infrastructure. In our scenario, an additional message between the proxy of domain A and the proxy of domain B would be needed in the standard SIP case (Fig. 1)

The JXTA-based location service has not yet been optimised. However, the time consumed of the first request can be optimised by periodically searching for newly published advertisements. Then, these advertisements are cashed locally, which apparently reduces discovery time.

6 Conclusion and Future Work

In this paper, we presented a novel concept for integrating P2P technologies into SIP-based multimedia networks. In contrast to former work, we build on

the standard and open source platform JXTA, which enables the integration of arbitrary routing mechanisms (allows customisation of routing mechanisms, especially on resource-limited devices). Thus, we provide a generic architecture for P2P-based SIP entities. The present measurements show the benefit of our approach. In general, the time consumed for sending messages from one UA to another one in a P2P-based SIP network is obviously lower than in a standard SIP-based network. As P2P-based SIP is especially useful in ad-hoc environments, and service location is a crucial part in these networks, we integrated our recently developed approach for SIP-only service location. This has the benefit that instead of using separate protocols, only one SIP stack is needed for session management and service location, respectively. SIP-only management reduces resource usage on small and mobile devices when implementing service registration/search in combination with capabilities registration/search. Classical SIP user agents can use standard SIP technology to access the virtual overlay network build upon JXTA. Using JXTA in combination with SIP to search for services in a JXTA-based P2P group in an ubiquitous environment provides better support for queries, as service descriptions are not standardised. In case of standardised service descriptions in adjacent environments, JXTA allows the integration of a DHT-based routing mechanism, which results in a better support for exact-match queries.

Service security is intended future work, as UAs should be able to trust their services. Therefore, registrations and modifications of service information have to be reliable. We will investigate certificate mechanisms for this issue [23], especially we would like to integrate these technologies with regard to P2P mechanisms. Finally, although we think that P2P SIP is especially useful for ad-hoc environments, we would like to evaluate our approach in a wide area network using the ns-2 network simulator.

Acknowledgment

We would like to thank Matthias Rink for the implementation of the very first prototype application.

The work described in this paper is based on results of IST FP6 Integrated Project DAIDALOS and of Project AKOM. DAIDALOS receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. AKOM is funded by the German Research Foundation, DFG.

References

1. Skype Limited. Skype. <http://www.skype.com>, 2006.

2. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. IETF RFC 3261, 2002.
3. 3GPP. IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP). TS 24.229 V7.4.0, Jun 2006.
4. D.A. Bryan, B.B. Lowekamp, and C. Jennings. A P2P Approach to SIP Registration and Resource Location. IETF draft-bryan-sipping-p2p-02, Mar 2006.
5. H. Schmidt, T. Guenkova-Luy, and F. J. Hauck. Service Location using the Session Initiation Protocol (SIP). In *IEEE ICNS'06*, Silicon Valley, USA, Jul 2006.
6. T. Guenkova-Luy, A. Schorr, F.J. Hauck, M. Gomez, C. Timmerer, I. Wolf, and A. Kassler. Advanced Multimedia Management - Control Model and Content Adaptation. In *IASTED EuroIMSA'06*, 2006.
7. T. Guenkova-Luy, A. Schorr, A. Kassler, I. Wolf, J.B. Blaya, T. Inzerilli, M. Gomez, and T. Mota. Multimedia Service Provisioning in Heterogeneous Wireless Networks. In *IPSI'05*, 2005.
8. D.A. Bryan, E. Shim, and B.B. Lowekamp. Use Cases for Peer-to-Peer Session Initiation Protocol. IETF draft-bryan-sipping-p2p-usecases-00, Nov 2005.
9. L. Gong. Industry Report: JXTA: A Network Programming Environment. *IEEE Internet Computing*, 5(3), 2001.
10. J. Eberspächer, R. Schollmeier, S. Zöls, and G. Kunzmann. Structured P2P Networks in Mobile and Fixed Environments. In *HET-NETs '04*, Ilkley, U.K., 2004.
11. J. Rosenberg and H. Schulzrinne. Session Initiation Protocol (SIP): Locating SIP Servers. IETF RFC 3263, 2002.
12. E. Guttman, C. Perkins, J. Veizades, and M. Day. Service Location Protocol, Version 2. IETF RFC 2608, 1999.
13. S. Berger, H. Schulzrinne, S. Sidiroglou, and X. Wu. Ubiquitous computing using sip. In *NOSSDAV*, pages 82–89, New York, NY, USA, 2003. ACM Press.
14. D. Willis Ed., D. Bryan, P. Matthews, and E. Shim. Concepts and Terminology for Peer to Peer SIP. IETF draft-willis-p2psip-concepts-00, Jun 2006.
15. S. Baset, H. Schulzrinne, E. Shim, and K. Dhara. Requirements for SIP-based Peer-to-Peer Internet Telephony. IETF draft-baset-sipping-p2preq-00, Oct 2005.
16. E. Shim, S. Narayanan, and G. Daley. An Architecture for Peer-to-Peer Session Initiation Protocol (P2P SIP). IETF draft-shim-sipping-p2p-arch-00, Feb 2006.
17. I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *ACM SIGCOMM'01*, pages 149–160, 2001.
18. A. Johnston and H. Sinnreich. SIP, P2P and Internet Communications. IETF draft-johnston-sipping-p2p-ipcom-02, Mar 2006.
19. K. Singh and H. Schulzrinne. Peer-to-peer Internet Telephony using SIP. In *NOSSDAV*, Skamania, Washington, Jun 2005.
20. The Internet Society. JXTA v2.0 Protocols Specification. Technical report, Sun Microsystems, Paolo Alto, USA, 2001, last changed 2005.
21. B. Traversat, A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J.-C. Hugly, E. Pouyoul, and B. Yeager. Project JXTA 2.0 Super-Peer Virtual Network. Technical report, 2003.
22. National Institute of Standards and Technology (NIST). Project IP telephony / VoIP. http://snad.ncsl.nist.gov/proj/iptel/, 2005.
23. C. Jennings and J. Peterson. Certificate Management Service for The Session Initiation Protocol (SIP). IETF draft-ietf-sipping-certs-02, 2005.

Teil II

Verteilte Anwendungen und Web Services

An Orchestrated Execution Environment for Hybrid Services

Sandford Bessler[1], Joachim Zeiss[1], Rene Gabner[1], Julia Gross[2]

[1] Telecommunications Research Center Vienna (ftw.),
[2] Kapsch CarrierCom

Abstract. Influenced by the success of web service technologies for the development and deployment of IT services, the telecommunications R&D community starts adapting and building similar service delivery platforms, trying to manage the hard constraints of telco domains, such as: existence of heterogeneous protocols, extensive use of asynchronous communications and real-time delivery requirements.
In this work we present a detailed analysis of an enhanced JAIN/SLEE architecture in which a workflow engine has been seamlessly integrated, being capable to orchestrate service entities from both telecom and IT domains.

1 Introduction

Given the intense competition in the telecommunication industry, the service providers have to decrease the time to market of new created services from eighteen to three months. In the IT world, techniques such as the web service composition have been increasingly used to create new services in a short time, even in a semi-automated way (see [3],[5] for composition approaches). In telecommunications, service development in the Intelligent Network environment has been busy for decades to solve the service or feature interaction problem: how to correctly react to a call event that would trigger several service features a user has subscribed to. With the apparition of new service delivery platforms for hybrid services such as Parlay or the IP Multimedia Subsystem, new entities to orchestrate services have been proposed such as Service Brokers (Parlay) or SCIM (Service Capability Interaction Manager in IMS).

In this paper we propose a novel service orchestration approach in a hybrid service environment that integrates IT resources and telco functionalities, trying to overcome obstacles like the existence of heterogeneous protocols, the extensive use of asynchronous methods (callbacks, events), and real-time delivery requirements.

The main innovation in our solution is to integrate a workflow engine using BPEL (business process execution language) inside a JAIN/SLEE (see section 2.2) container. In this way, we create an orchestrated service environment that uses fast deployable BPEL scripts to control service building blocks or external entities via a variety of protocols such as SIP (session initiation protocol), INAP (intelligent network application part) or SOAP. The performance increase compared to the standard composition of web services stems from the low latency in the communication between internal service components. Much flexibility is gained as well because, from the perspective of the service composition engine, it makes no difference whether the orchestrated parties are external web services or service building blocks running locally in the SLEE container. In contrast to that, a BPEL engine running externally to the SLEE container would use either SOAP or a remote protocol and would need converters to connect to SLEE service building blocks.

A common drawback of heterogeneous platforms is that the service logic is distributed and it is difficult to package it in one single deployable unit. Our approach

makes this possible that internal SBBs and BPEL scripts can be deployed together. Third party service enablers remain external but are represented as business process partners in the deployed scripts.

To our knowledge, no work published at the time of writing has conceived the proposed integrated architecture using a full standardized environment such as JAIN/SLEE with a standardized process scripting language such as BPEL, to process hybrid (Telco-IT) services.

In the literature we find quite a few works devoted to the orchestration of telco services: the authors in [4] present a service architecture comprising an orchestration function using BPEL, without mentioning the use of a standardized platform. All the service components expose web service interfaces. In [2] the authors consider a SLEE environment (StarSLEE) as well and analyze composition using BPEL, but they seem to reject the idea of integrating an existing BPEL engine.

Our orchestrated service platform can be optimally deployed in an IP Multimedia Subsystem (IMS) service context. Although other service platforms exist for integrating SIP, IN and web based protocols (such as OSA/Parlay and Parlay X), the native SIP application servers provide more flexibility. Thus, they can process a SIP message with extended headers or body information without having to fit it to a certain service capability function like call control, presence or messaging. Indeed, a SIP message from a user client can convey additional information to start or control a service application; therefore, any server on the signalling path has to make sure that this information can reach the application. Among native SIP application servers, an alternative solution is based on the SIP servlet API, which uses the well known programming model of HTTP servlets. We decided however to use the JAIN/SLEE technology because of the flexible resource adaptor concept that can be applied to other protocols than SIP or SOAP, and because of the powerful event model, as we will see in the architecture section.

1.1 A motivating example

A real time scenario that demonstrates the service composition benefits is called "Expert on Call" and has been described in [1] in the context of an IMS environment. In this scenario, a technician is debugging a problem somewhere on site. In case he encounters difficulties in the process of solving the problem, he activates the "Expert on Call" application and asks advice either from a colleague who is working nearby or from one of the expert group back in the office. Before putting the call through to a colleague nearby (using a location service (LS) to determine who is nearby) or to an expert, the application determines the availability information of the potential helpers (using the presence service, PS). In the "Expert on Call" the service has to be provided in real time, i.e. in one and the same service call initiated by the technician. Once a voice connection has been established, other services can be added to the session: video and/or file transfer to help to eliminate the problem, voice/video/data conferencing with several colleagues or experts, etc.

The rest of the paper is organized as follows: in the next section we describe shortly the SIP, SLEE and BPEL technologies, then, in the main section 3, the proposed architecture and the basic interactions are introduced. Based on the example, we illustrate the operation of the system in section 4, present preliminary performance results and discuss a number of lessons learned from using the technologies, in section 5.

2 Technological environment

In this section we introduce the main technologies needed to build the orchestrated service environment.

2.1 SIP

The Session Initiation Protocol is an IETF standard [8] used in telecommunication services to control multimedia sessions. SIP has been adopted by the 3GPP in the definition of the IP Multimedia Subsystem(IMS), a SIP overlay network on top of the UMTS packet domain. Therefore, SIP is an important protocol which will be implemented in most mobile phones and which will be the main way to asynchronously notify a user about an event or a message, replacing SMS and WAP push. Besides signalling and preparing other protocol channels to be used by the terminal application via HTTP for example, SIP can convey payload in its messages (INVITE, OK, SUBSCRIBE, NOTIFY, MESSAGE) and communicate in this way with the service without the need of another application protocol (at least for simple applications).

2.2 JAIN/SLEE

JAIN SLEE [9] is the Java standard for a Service Logic Execution Environment (SLEE), which is a known concept in the telecommunication industry to denote a low latency, high throughput event processing environment. Standardized by the Java Community Process JSR22 in 2004, the SLEE architecture emerged from the J2EE container in which Enterprise Java Beans perform application logic, however the new SLEE container and the components had to be adapted to the requirements of telecom services: support of asynchronous events, low latency, support of heterogeneous protocols such as SIP, INAP, SOAP, etc. The resulted SLEE entities as shown in Fig. 1 below are:

- Service Building Blocks (SBBs), software components in the sense of EJBs, which can be cascaded to build hierarchical structures and support event communication,
- Resource Adaptors (RAs) that conform a certain resource adaptor type and map a certain protocol or resource actions to SLEE events. Specifically the communication from RA to SBB is synchronous, whereas communication from RA to SBB is event based.
- Activity and Context objects are needed to share session state information between SBB and RA across multiple events. This represents a state machine, a typical requirement for most telecom applications.

2.3 BPEL

The Business Process Execution Language (BPEL)[10] is an XML construct for describing business process behaviour. BPEL is layered on top of other Web technologies such as WSDL 1.1, XML Schema 1.0, XPath 1.0, and WS Addressing. The BPEL notation includes flow control, variables, concurrent execution, input and output, transaction scoping/compensation and error handling.

Business Processes described with BPEL usually use external Web services to perform functional tasks. Since in our case the BPEL process is inside the SLEE platform, it has to be able to fire JAIN/SLEE events to the SBBs and RAs. These web services are described by WSDL files, refer to partners, and their link types. The WSDL files describe the structure of messages (data containers), operations (method calls), and port types (operations and input, output, and error messages). They also bind port types to particular protocols such as SOAP, operation call methods such as RPC, or in our case the definitions of JAIN/SLEE events.

There are several reasons to select BPEL as orchestration language:

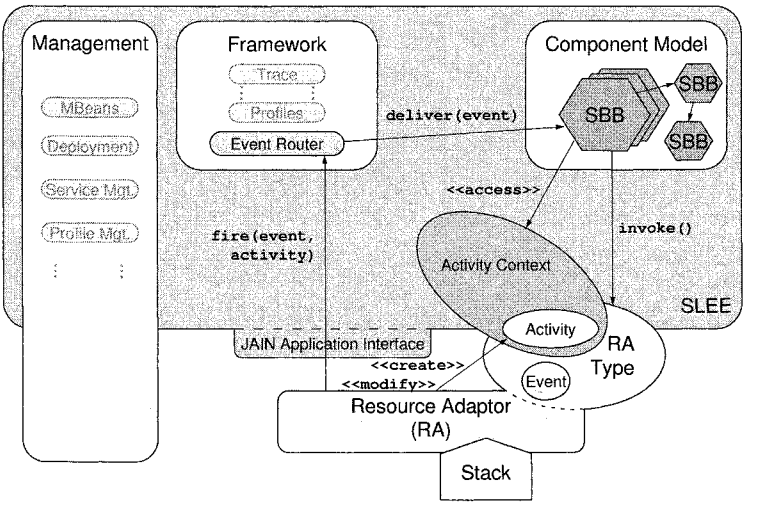


Fig. 1. Main Entities of a JAIN/SLEE service environment

- It offers a generic representation of a business process (possible to re-deploy on different platforms).
- It offers high level abstraction (programming in large) for rapid process creation. It therefore allows network operators, sales person or even subscribers to define small business processes. Existing visual BPEL design tools simplify this task.
- A scripting language offers easy development, maintenance and adaptation of a business process (no need to re-compile the software in case of changes).
- Its underlying infrastructure supports exception handling, transactions, compensation mechanisms, service selection and binding, etc. These functionalities are supported by BPEL and are normally provided by the composition engine.

3 Architecture of the orchestrated service execution environment

Having shortly reviewed the main technologies involved in the execution environment, we proceed with the proposed integrated architecture comprising a SLEE container with SIP and IN resource adaptors, a new kind of resource adaptor hosting the BPEL engine, and a number of service building blocks (SBBs).

In order to integrate the BPEL engine in a SLEE container, we follow the paradigms of the SLEE architecture. Therefore, the BPEL engine is wrapped in a resource adaptor (BPEL RA). When deployed into the SLEE, the resource adaptor defines the events it might fire, as well as a resource adaptor type interface that can be called synchronously by SBBs. The communication from SBB to BPEL RA is done via synchronous method calls, whereas in direction from BPEL RA to SBB, the resource adaptor fires events (see Fig. 1). The BPEL RA starts a servlet container which has the BPEL engine deployed. Into this engine any BPEL process script can be deployed and accessed from inside the container via local Java calls or from outside the container via SOAP. In the following description we assume for the sake of simplicity that the service logic in form of BPEL scripts is triggered by a

SIP or an IN message, however the general case allows any SOAP or internal event to do the same.

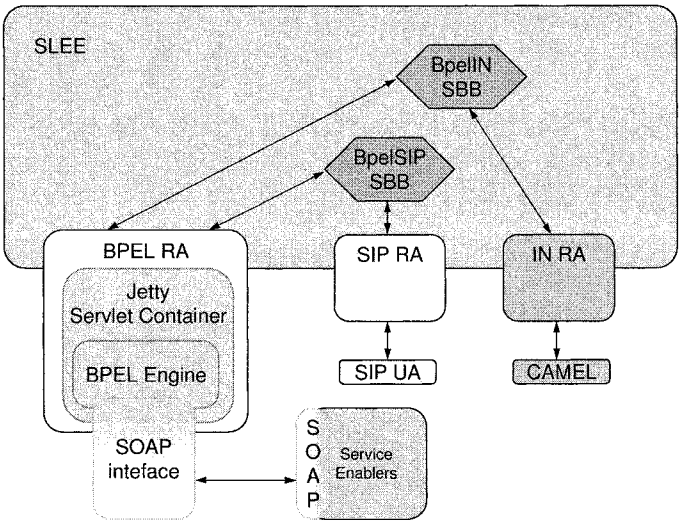


Fig. 2. High level architecture of the converged execution environment

Fig. 2 shows that the BPEL RA, which, like any other resource adaptor, can be accessed directly only by SBBs, but not by other resource adaptors. Therefore, to allow communication with the BPEL script, we need to define an SBB that handles protocol events on one side and BPEL script calls and response events on the other. SBBs that want to act as partners of a BPEL process need to listen to and process request events coming from the RA. Whether a partner is an SBB inside or a conventional web service outside the SLEE is defined in the process deployment descriptor file.

JAIN/SLEE events are associated to activities (see Fig. 3) which in turn are manipulated by ActivityContext objects. In order to invoke a BPEL process, it is called with the process/script name and its parameters, then a new activity is created by the BPEL RA. The result of this request is delivered always as an asynchronous event to the calling SBB. This event is fired on the same activity which was created for the request. After firing the response event the activity ends and the resources needed for handling the script invocation are freed.

The JAIN/SLEE event router transports events associated to activities, which helps to address interested SBBs and RAs. Furthermore, the activities keep state of the actual transaction. For the BPEL RA this means that we need to assign an activity to each BPEL process invocation regardless of the request direction (from SBB to RA or vice versa). The mapping between activity and process id is the glue between SBBs and BPEL scripts.

As the integrated BPEL engine is deployed into a servlet container, it is in contact with the outside world. Any of the BPEL processes deployed in the engine are available as web services. Internal SBBs and external web services can be arbitrarily combined to make a complex service. One can use this concept to interweave

internal and external components, services or applications in a single BPEL script. There is no need to maintain heterogeneous distributed server farms. This is because JAIN/SLEE components and applications requiring remote service enablers, web services or telecom protocol stacks can be deployed in one spot together with the combining orchestration script. Communication services written as a BPEL process inside the SLEE are agnostic to the transport protocol.

As depicted in Fig. 3, the communication between SBBs and BPEL scripts is done using request and response objects inside method calls or via event delivery. To communicate across class loader boundaries these request and response objects are simple Java Strings in form of XML snippets which contain the request or response parameters.

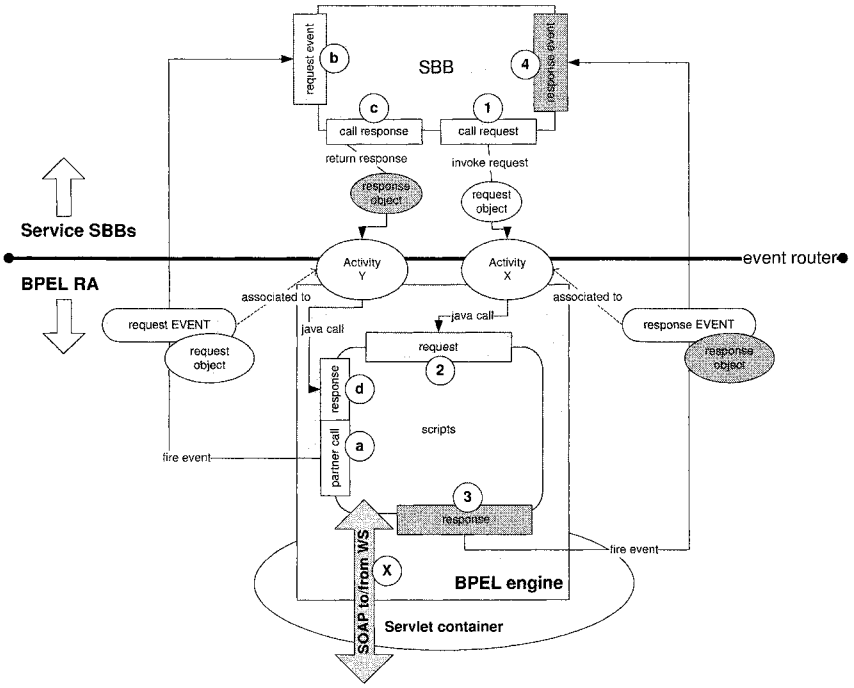


Fig. 3. BPEL Resource adaptor architecture and interactions

3.1 Basic interactions

In Fig. 3, we show in more detail the interactions of components inside the orchestrated environment. The two main players are the BPEL engine wrapped in a resource adaptor, and the proxy SBB that is responsible for the communication with the SIP RA (not visible in the figure).

For an SBB initiated process invocation we present the following message walkthrough (the numbers correspond to those in Fig. 3):

(1) The SBB creates a request object containing the script name and parameters; it asks for a new activity and calls the invoke request method on the BPEL RA; the call is returned immediately.

(2) The request is injected via a local java method invocation to the BPEL engine; it invokes the script in asynchronous mode given the script name and parameters; the BPEL process ID is stored against the activity.
(3) When the process is finished, a response object containing the script result is created; this object is then fired in a response event belonging to the activity stored against the BPEL process ID.
(4) The SBB receives an event for a particular activity. This activity is most likely linked to another activity coming from the network (e.g. SIP call); based on those two activities, the SBB takes further action; the activity ends.

Similarly we show the walkthrough for a BPEL process initiated partner call to a service SBB inside the application server (the letters correspond to those in Fig. 3):

(a) The script calls a partner which is deployed to be invoked locally; if an activity exists for that process, it is used, otherwise a new activity is created; a request event is fired on the activity.
(b) The SBB receives the request event.
(c) The SBB calls a response method based on the request name; the SBB decides to take action or not; when processing is finished, it invokes the response method on the BPEL RA providing the response object, the requests result and the related activity.
(d) The response object is forwarded to the process/script associated to the given activity; the activity ends and the processing resources are freed.

4 The example scenario

This section discusses the end-to-end message and process flow of the expert-on-call (EOC) service mentioned in Section 1, using BPEL scripts deployed in our BPEL resource adaptor. The described service operates in an IMS context, with the SLEE environment as an IMS application server, however the orchestrated environment can be deployed for any hybrid or converged service as well. The mentioned external Location, Presence and Group Management services may be offered by service providers or network operators, different from the provider of the EOC service.

The scenario in Fig. 4 starts with a SIP enabled terminal calling the EOC service. The called number (Request URI) determines the expert group, e.g. network-expert, application- software-expert, etc. This address determines the group list to look up at the Group Management service enabler. The location and presence status of all group members is obtained (by calling the service enablers mentioned above) and is compared to the location of the originator. Depending on the above information a (prioritized) list of terminating addresses (i.e. the experts) is made available.

4.1 Detailed service walkthrough

For the following service walkthrough we consider SIP terminals, however, functionality inside the BPEL script can be used for IN calls as well (the numbers in brackets refer to the call flow steps depicted in Fig. 5):

Part 1: Protocol Stack, pre-orchestration phase

- The SIP RA receives an INVITE with request URI:sip:experts@eoc.ims.net (1).
- The SIP RA is provisioned to identify the SBB to be informed (proxy SBB), based on the URI (2).

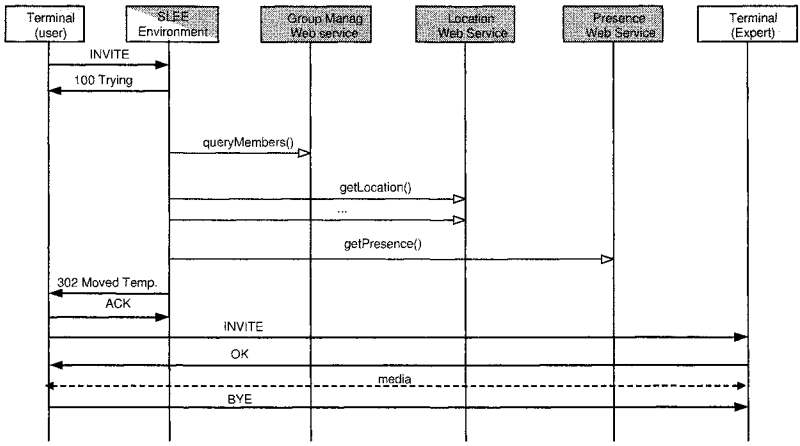


Fig. 4. Expert on call flow

- The SIP RA triggers TRYING response to be sent to originator (omitted from Fig. 5))
- The Proxy SBB identifies the script corresponding to the EOC service, creates new Activity and calls the BPEL RA with BPEL script URI and parameters (3).

Part 2: BPEL Script invocation orchestration phase

- The BPEL RA gets called from proxy SBB (3), takes script URL and parameters and uses the two to inject a web service request into the BPEL engine (4). The RA returns immediately.
- The BPEL engine finds the SIP frame script based on URL and feeds it with the parameters (5). Based on these (SIP specific parameters) the frame script decides which service script to call; in our case the EOC script. The related BPEL process ID is stored against the JAIN/SLEE activity in order to fire the response event to the correct requestor instance.
- The EOC script needs the names of the expert-group and of the caller, both taken from the SIP request.
- The EOC asks Group Management partner to retrieve the list of experts (6a).
- The EOC asks Location Service partner to locate the caller (6b).
- The EOC asks Location Service partner for each expert's location and validates proximity to caller (6c).
- The EOC checks for all the experts nearby their availability by asking Presence service
- The BPEL engine returns the list of available experts and the action "Forward-Call" to take effect on protocol side (7).
- The callback method inside the BPEL engine gets the EOC response and forwards it to the BPEL RA (8).
- The BPEL RA maps response to activity, puts the response data in generic BPEL response event, similar to what the proxy SBB did for the request, and fires the event on the corresponding activity, the proxy SBB receives the event (9)

Part 3: Protocol stack, post-orchestration phase

- In the previous step, the proxy SBB received the BPEL response event. The event contains response data in form of a XML snippet conveying the BPEL scripts response information.
- The "ForwardCall" action will now do the following: answer the INVITE message with a "302 moved temporarily" message (see Fig. 5) redirecting the call to the address of the selected expert: (10) invokes the SIP RA (11) sends the message.

The interface (i1) in Fig. 5 depicts the boundary between SBB and RA. The internal interface (i2) depends on the selected BPEL engine.

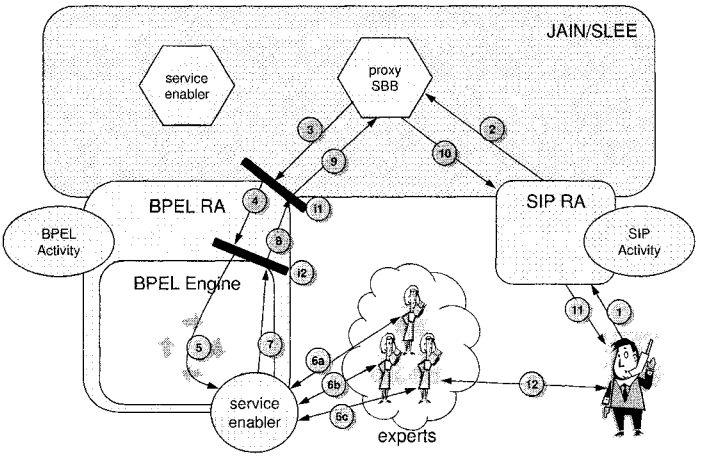


Fig. 5. Expert on call flow

5 Results and lessons learned

5.1 Performance Measurements

We implemented a prototype of the system containing the BPEL RA with an integrated BPEL engine and the necessary SBBs. The service script corresponding to the Expert on Call example has been written and deployed. To get an idea about the time spent in different parts of the system, two different scripts have been tested (see Fig. 5): a) the BPEL echo script in which the call is redirected to the BPEL engine returning a redirection address without the invocation of external WS partners and b) the full EOC script in which location, group management and presence services are invoked by the BPEL engine. We measured the call setup times of a sequence of 1000 calls between the SIP INVITE and 302 MOVED TEMPORARILY messages.

The tests have been performed on a Pentium 4, 3.2 GHz system with 2.56GByte RAM, under Suse 9.2. This machine was hosting the OpenCloud Rhino 1.4.2 SLEE container [14]. Partner web services have been deployed on a Pentium 4, 3.2 GHz system with 512MByte RAM. The three external web services used standardized Parlay X web service interfaces (and run inside an axis container version 1.3), however they didn't call any core components, but returned immediately with predefined values. Client SIP messages have been generated on a separate machine with SIPp test tool version 1.0 [15].

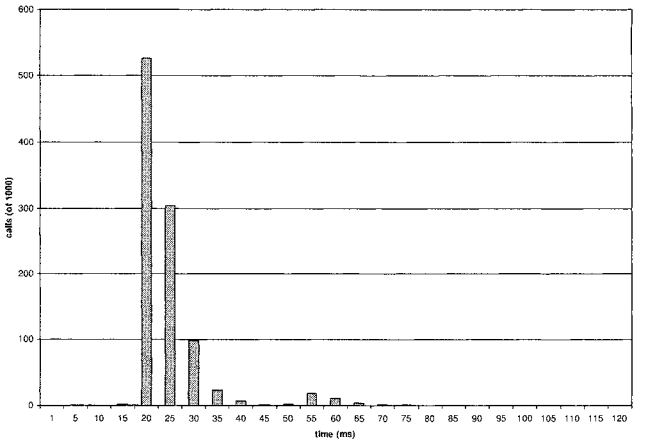


Fig. 6. Distribution of service execution times for an ECHO script

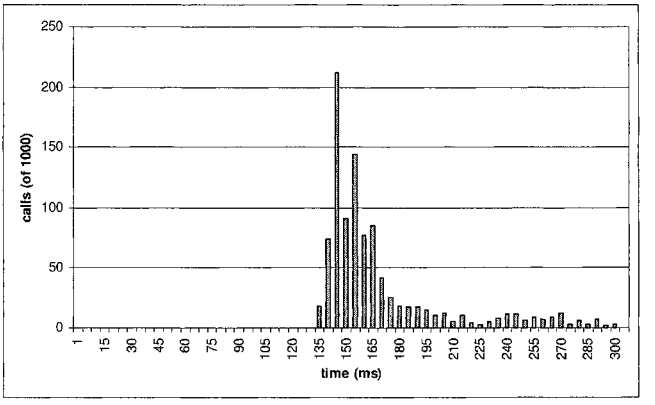


Fig. 7. Distribution of service execution times for the EOC service

The preliminary tests show for the Echo script a mean call setup time of 24ms and for EOC script a mean call setup time 173ms. The setup time distributions of

1000 experiments can be seen in Fig. 6 and Fig. 7. Note the long tail distribution in the EOC case, due mainly to the interactions with external web services.

Taking a closer look at the implementation of our prototype, we found out, that the major part of time needed for the EoC script to complete is spent on external web service partners invocations. The longest period of approximately 80-90 ms is spent for the location service invocation (according to ActiveBpel process logs). The group management service needs around 40 ms and the presence service needs 15 ms to complete. The relatively long delay for the location service invocation can be explained by the high complexity of the SOAP interface (both nested and verbose) described by the Parlay-X API [6]. Other process activities, such as initialization of complex process variables and array iteration loops take around 15 ms. Significant performance improvements could be achieved by using a BPEL engine designed for production use and optimizing the BPEL RA implementation. Given the prematurity of our implementation and the usage of open source s/w components instead of production quality BPEL systems, the results obtained encourage us to believe that our approach can be used for creation of hybrid telecom services. The next steps are to further improve implementation regarding concurrency handling and to run load tests on a production system.

5.2 Lessons learned

The initial release of the JAIN/SLEE standard (version 1.0 JSR 22) faced the problem of proprietary resource adaptors (RA) made available by SLEE vendors. This fact tied the services to the application server product used for their development. As the required RA could only be deployed to a particular SLEE implementation, the service using the RA inherited the same restriction. With version 1.1 (JSR240) this problem is intended to be solved, as a number of vital resource adaptor types are standardized. However, there are no standardized implementation guidelines for a RA. Therefore, while interfaces are compatible, implementation details could still tie a RA implementation to a particular SLEE vendor. Our implementation of the BPEL RA had to be tailored to the specific needs of the application servers class loading strategy. As a result, our resource adaptor implementation would still need to be adapted for different SLEEs even if the interface is standardized.

Furthermore, sending a request from an SBB to a BPEL process implies passing information across class loader domains. As a result, request and response parameters cannot be delivered in typed objects, but only in XML structured Java strings. This fact significantly worsens the real time performance of the prototype. A better integration into the JAIN/SLEE container is necessary for further real time improvements. If the BPEL engine would be tightly coupled to the event router of the system, script invocation would be much faster and event routing itself could be performed via BPEL scripting. Designing BPEL processes is easy if the partner interfaces are reasonably simple. If complex nested data structures are used as input/output variables, the data handling can become increasingly non-trivial, especially if arrays are used. Visual tools and BPEL process simulators proved to be extremely useful for designing and debugging BPEL processes.

In the process of implementing the prototype we have encountered two models for deploying and invoking partners as java classes, therefore allowing local java calls to those partners. The first model using WSIF (Web Services Invocation Framework) [13], allows defining new WSDL bindings, where java binding is one of many possible (the most spread binding is http binding). How the WSDL service is deployed and what transport it uses, is defined in the WSDL document. All the transport details are transparent to the process. The second model exploits a process deployment descriptor to indicate those partners that are to be invoked locally and those - per HTTP/SOAP. The partner implementation that is to be deployed as Java class

needs to implement a certain invoker interface and has to parse incoming XML messages to build expected input Java method parameters. For reasons of quick prototype implementation we integrated an open source BPEL engine [12] that offers this option. The first (WSIF) model, however, is much more generic and provides a neat clean solution to multi transport support which is both transparent to the BPEL process and the partner implementation.

6 Concluding Remarks

This paper presents a novel architecture and design considerations for a service orchestration environment suitable for telecom and internet applications. The early measurements show that we have achieved an efficient integration of the BPEL orchestration engine into a Jainn/SLEE environment, however the invocation of more complex external web services remains time-consuming and has to be considered in the service design phase.

In a further task we would investigate, how our orchestration approach speeds up the time needed to develop new services using both SIP and IN interfaces. In order to achieve fast service development, we have however to consider all the system components, including a terminal development platform for SIP applications.

7 Acknowledgement

This work has been funded by the Austrian Government Kplus program. The authors would like to thank their colleagues in the SIMS project and to Charlie Crighton from OpenCloud for fruitful discussions.

References

1. J.G. Adamek, E.H. Henrikson, H.A. Lassers, A.Y. Lee, and R.B. Martin, Services and Technical Considerations for the Wireless IP Multimedia Subsystem, BLTJ vol.7, Issue 2, 2002
2. Baravaglio, C.A.Licciardi, C. Venezia, Web Service Applicability in Telecommunication Service Platforms, International Journal of Web Services Practices, Vol 1, No.1-2, 2005
3. B. Srivastava, J. Koehler, Web Service Composition - Current Solutions and Open Problems, in Proceedings ICAPS 2003
4. T.Pollet, G. Maas, J.Marien, A. Wambecq, Telecom Services Delivery in a SOA, Proceedings of Advanced Information, Networking and Applications,AINA06, p.529-533
5. N. Milanovic and M. Malek, Current Solutions for Web Service Composition, IEEE Internet Computing, Nov-Dec. 2004
6. Parlay, http://www.parlay.org
7. The IP Multimedia Subsystem, G. Camarillo, M.A. Garcia-Martin, Willey 2005
8. Session Initiation Protocol, IETF RFC 3261, http://www.ietf.org/rfc/rfc3261.txt
9. JAIN SLEE API Specification, Sun, Java Community Process, JCR 22
10. Business Process Execution Language for Web Services , IBM, http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
11. T. Oh-ishi, T. Iwata, S. Tokumoto and N. Shimamoto, Network Services using Service-Composition Technology, Proceedings Networks 2004, Vienna, Austria
12. ActiveBpel, http://www.activebpel.org
13. Web Services Invocation Framework, http://ws.apache.org/wsif
14. OpenCloud Rhino, http://www.opencloud.com
15. SIPp test tool, http://sipp.sourceforge.net

A Lightweight Service Grid Based on Web Services and Peer-to-Peer

Markus Hillenbrand, Joachim Götze, Ge Zhang, and Paul Müller

University of Kaiserslautern, Germany
Department of Computer Science
{hillenbr, j_goetze, gezhang, pmueller}@informatik.uni-kl.de

Abstract What is *The Grid*? This question has been asked by many people and has been answered by many more. On our way to the One Grid, it is currently only possible to distinguish between several Computational, Data and Service Grids. This paper introduces Venice, a lightweight Service Grid that allows for easy service deployment, easy maintenance and easy service usage. It contains management services, information services and application services that can be used to build distributed applications upon. Its main focus in on providing a flexible and dependable infrastructure for deploying services that do not require special knowledge and expertise in Grid computing.

1 Introduction

In 1969 Leonard Kleinrock imagined "the spread of computer utilities, which, like present electric and telephone utilities, will service individual homes and offices across the country" [1]. Ian Foster and Carl Kesselmann in 1998 initially tried to define a Grid that could help implement such computer utilities [2]. After several iterations, Ian Foster finally published his three point checklist [1] that must be met by a system to be called a Grid.

Today, a distinction between *Compute Grids*, *Data Grids* and *Service Grids* can be made to distinguish between the Grid systems available. There is no common definition for these Grid types, but here they are understood as follows. A Compute Grid provides resources for high throughput computing and makes possible time consuming calculations over the Internet (e.g. by giving standardized access to clusters). A Data Grid makes available disk space to securely store large data sets. A Service Grid finally uses these Grid types as a commodity and adds more value to the end-user by providing special services like virtualized applications or accurate algorithms and offering intuitive graphical interfaces to access those services.

A lightweight Service Grid should additionally be *easy to deploy* (i.e. service developers have the ability to create and deploy services very quickly and with tools provided by the Service Grid), *easy to maintain* (i.e. service providers get tool support for managing, monitoring and configuring services at runtime) and *easy to use* (i.e. service consumers have the ability to use the services of the Service Grid without having to install special software and can access service functionality with graphical user interfaces). A Service Grid should be built on open standards and be applicable to different usage scenarios – not only high performance computing or high throughput data processing.

While Compute Grid like Unicore [3], gLite [4] and Globus Toolkit [5] (GT4) as well as Data Grids like the EU DataGrid Project [6] are deployed on a larger scale and used in world wide research projects, pure Service Grids are not yet available. GT4 could be regarded as an early version of a Service Grid, but its legacy (especially its Grid Security Infrastructure and the pre Web services components) makes it neither lightweight nor easy to deploy, use or maintain.

In the following, the Venice Service Grid will be outlined on a top level view. Some details about several services have been published previously, but they have since then been reorganized and completed to create a Service Grid. The roots of the Venice Service Grid are in a telephony project [7] that successfully made available signaling protocols like SIP, H.323 and several dozen supplementary services as Web services. In section 2 the overall architecture is explained and the services available in Venice are introduced. A performance measurement of the basic use case for accessing a service is shown in section 3 while section 4 concludes the paper in gives an outlook.

2 Architecture

The Venice Service Grid is an open framework for distributed applications that allows the easy creation, deployment, integration, and usage of services. The means to achieve this are virtualization of the resources used and abstraction from the underlying technology using services. By using the Venice Service Grid it is possible for a service provider to run a single security domain with services for the users of his domain. But it is also possible to connect other domains in order to gain access to the services provided by these domains. Such a service federation gives seamless access to all authorized services within the service federation. Every provider is primarily responsible for his own customers and services, and Venice provides a secure and standardized passage for data exchange and service access over the Internet using Web services and Peer-to-Peer (P2P) technology (as illustrated in Fig. 1).

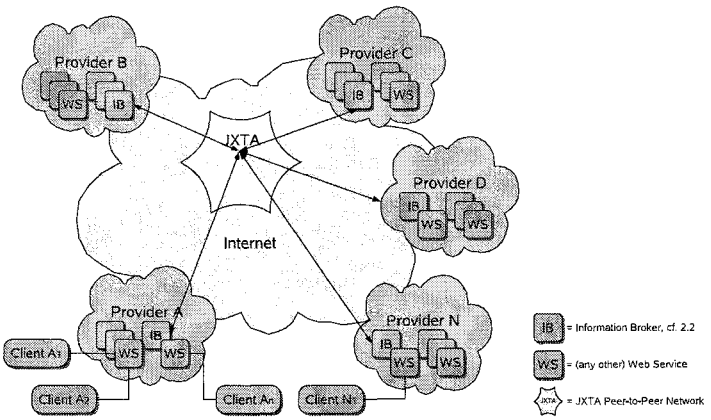


Figure 1. The Providers' Perspective

The services offered by the Venice Service Grid are implemented as Web services directly on top of the Tomcat [8] servlet container and the Axis [9] SOAP engine. The services are described using the Web Services Description Language (WSDL) and communicate with SOAP over HTTP. The interface definitions are separated in an abstract service interface definition containing the port types and messages, a concrete interface definition containing the concrete binding, and a concrete service implementation definition containing the ports and locations of the real implementation. All data exchanged between the service entities are specified in publically available XML schema files. This allows for re-use of the WSDL and XML schema files in the service descriptions.

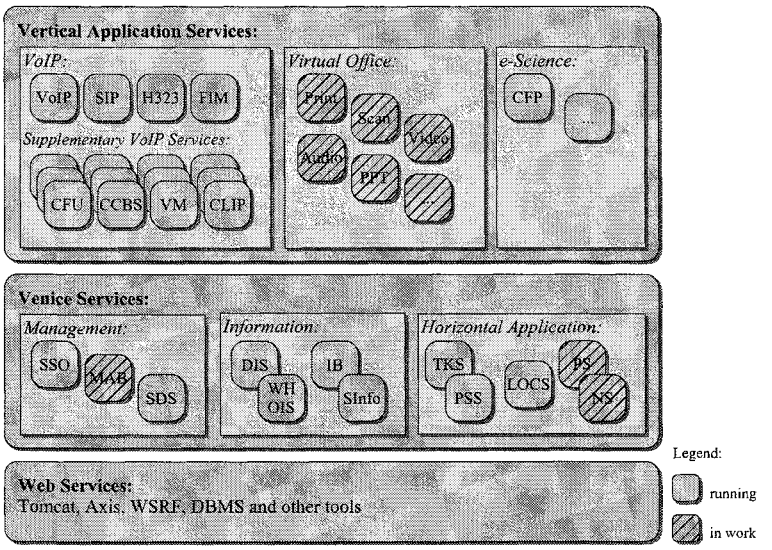


Figure 2. Architectural Overview

The services of the Venice Service Grid can be divided into three distinct categories (cf. 'Venice Services' in Fig. 2). *Management Services* are services necessary for the common usage and maintenance of users, resources, services, and other entities in the Grid. The *Information Services* are responsible for collecting, managing and distributing data needed by users or services. The *Application Services* finally are horizontal services that can be used by any application built on top of the framework and vertical services that are mostly valuable in a specific application domain. An application consisting of several vertical services ties together the Management, Information and horizontal Application Services to predefined user accessible workflows. Common to all those services is that they are equipped with at least one graphical user interface that can be started using a special software deployment service (cf. 2.1). Thus, every service can also be used separately.

In the following subsections, the services already developed and currently being developed will be briefly introduced.

2.1 Management Services

In order to create an open and dependable service-oriented system, several basic services have to be provided that deal with management and maintenance of all services within the system. The following management services are part of Venice and make available authentication, authorization, accounting, billing and software deployment.

Single Sign-on Service (SSO). In order to access all services of a provider, a token-based single sign-on strategy [10] is the most promising approach for a lightweight Service Grid. A user has to authenticate once and will then receive a token allowing him to proof his identity in any further communication by providing this token. If the user is accessing a service, this token will allow the user not only to proof his identity, but also to proof his right to access this service. As there may be several services involved in a usage scenario, it is very important to use an authentication and authorization strategy based on single sign-on. This strategy allows the user to be authenticated to any service inside the authentication domain without the need to enter his credentials a second time.

The process of authentication in the Venice Service Grid is closely related to the authentication protocol Kerberos [11], but it is extended to fulfill the needs of a federated SSO environment on the basis of Web services. The Kerberos Protocol has the advantage of providing an easy to use basis for a SSO environment utilizing the security token to prove a successful sign-on.

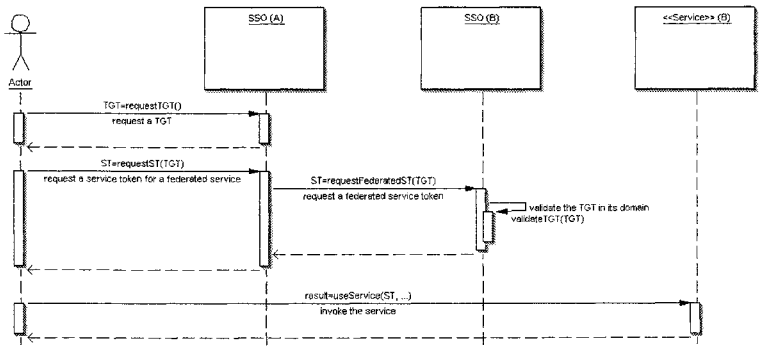


Figure 3. Authentication and Authorization for a Federated Service

An authentication and authorization infrastructure consists of four basic components. The first component is the user or client. This component has to be integrated in the system of the client and handles any operation needed for authentication and authorization. The authentication service – providing everything needed to allow a user to claim an identity and proof that this claim is justified – is the second component. If the user has successfully claimed an identity, he receives a token (TGT) to prove his identity. This TGT has to be used with the third component: the authorization service. This service provides the user with service tokens (ST) that have a limited life-time

and allow him access to services, if he has the privileges to use these services. Finally there is the service component. The task of this component is to ensure that the user of the service can provide a proper ST, i.e. he is successfully authenticated and received authorization to access the service, and that the access rights are sufficient to utilize the requested service functionality.

In the Venice Service Grid, the authentication and authorization functionality is bundled in the SSO service. The Venice Service Grid offers a distributed authorization management. Every service is responsible for providing and interpreting authorization information. Therefore, the SSO service requests from a service the possible authorization attributes and their range. The service provider can then assign authorization information to user roles. This data will then be encoded by the originating service and stored into the SSO's database where it will be used to create service tokens. Also the SSO service can handle different types of service requests. A user signing on can belong to the local authentication domain or to a federated domain. Additionally, a user can call local services or services provided by a federated domain. A local user calling a federated service is illustrated in Fig. 3. Combined with the distributed authorization management a flexible cross-domain authentication and authorization infrastructure can be provided. A more detailed description of this token-based single sign-on solution developed for the Venice Service Grid can be found in [12].

Metering, Accounting, and Billing Services (MAB). It lies within the nature of a service-oriented architecture with a sophisticated authorization management to offer services that provide the same functionality with different quality of service for different users – e.g. according to their authorization level. And these services can be offered by different autonomous service providers. In the end, a flexible and open mechanism for metering and accounting service usage has to be provided as an infrastructural service. On top of that, a transparent and secure billing has to be guaranteed. The Venice MAB services provides an accounting and billing framework for measuring Web services, collecting and processing Web service usage records, generating corresponding accounting information, and charging and billing for the Web services usage according to the billing strategies. The MAB consists of four layers: the probe layer, the metering layer, the accounting layer, and the billing layer.

The *probe layer* comprises different types of probes which are responsible for measuring different types of Web services or other objects. These probes can either be integrated into the Web service or in the host where the Web service is running [13]. Different measured objects and different metrics require different probes; e.g. probes for measuring the duration of a Web service execution or probes for measuring the CPU utilization of a Web service instance. The *metering layer* provides services for gathering measurement information from probes, organizing, storing and transferring the measured data, and managing the behavior of probes according to accounting policies. The meter data is stored in meters temporarily and must be kept until it is collected by an accounting service. The *accounting layer* provides services for gathering meter data from different meters in its administrative domain, processing the meter data to generate Usage Records (UR) according to the accounting policies, and sending the URs to a billing service. Accounting information about composite Web services will be

correlated in this layer [14]. The accounting service is also responsible for controlling the behavior of the meters. URs generated by the accounting service are stored in an accounting information database. The URs stored in this database should be permanent and must be stored securely against unauthorized access. In a multi provider scenario, a token based accounting using P2P technology (like [15]) is under investigation. The *billing layer* provides services for collecting the URs from the accounting services, for calculating the resource usage for different users according to the pricing scheme and for generating the billing records. The billing records can be used to generate different reports. Users can then retrieve their billing reports. If a certain resource usage provision must be met, notifications will be sent to the authorization service to control different users' resource consumption.

Software Deployment Service (SDS). In order to provide an easy-to-use application to the end-user, it is important to dispense the user from tasks like installing and/or updating software. But software deployment and installation is a tedious task. In a service-oriented (or client / server) environment, a lot of different client and server types may participate. Clients have different kinds of hardware and software prerequisites they bring along while trying to access services. Notebooks and desktop computers are built using completely different processor architectures and operating systems than PDAs or cell phones provide. But within a service-oriented architecture, all those clients should be able to access and use the same service, regardless of their capabilities and prerequisites.

The Venice Software Deployment Service allows the user to run an up to date application without any further effort if his client is supported by the application provider. For most of the Venice services, this client software is just a graphical user interface and the client libraries needed to access Venice services. Additionally, such a service enables a service provider not only to maintain a state of the art software infrastructure, but also to automatically and dynamically replicate certain services to ensure service availability. This compensates for high load situations and establishes a replica management. As a result the Software Deployment Service reduces the administration effort on both sides, the user and the service provider. A comprehensive overview of the Software Deployment Service can be found in [16].

2.2 Information Services

A Grid needs a starting point where all necessary data can be found to access all other services. In the Venice Service Grid, the *Domain Information Service* provides all that data to users and services. Additional information services provide data about users and services or make local data available to other service domains.

Domain Information Service (DIS). This service is responsible for providing all necessary information for using services in a domain. Thus, the service is the starting point for any further service interaction and for example provides meta data about the single sign-on service, the information broker and the service domain itself.

Who Is Service (WHOIS). Using this service, it is possible to retrieve data about other users in the local or a remote domain. The data accessible through this service has to be authorized by the affected user in order to respect the user's privacy. This service will be complemented by a presence service in the future.

Information Broker (IB). This service is a generic service responsible for brokering service meta data, user meta data and other arbitrary data sets. The service meta data is used to find a suitable service for the user and can contain various kinds of data, e.g. data about the interface, dependability information, user's ratings about usability, etc. The user meta data is ranging from basic status information, e.g. the online status of the user, to more sophisticated data like the personal details of the user. These personal details may include textual information, but also other data is possible, e.g. audio files, images, etc. The implementation of the Information Broker is based on the JXTA P2P technology in order to attain scalability and to overcome some shortcomings of UDDI and other central registries [17].

Some of these shortcomings are *centralization* and *uptodateness*. A directory service like UDDI is typically a centralized service providing data about the registered services. This results in a single point of failure if the service is not available. During operation a single service can be a performance bottleneck and account for a severe performance impact. In Venice, the P2P technology is used to implement a decentralized registry. Additionally, the quality of a directory service is closely related to the quality of the data it is providing. In most directory services the data is entered manually and therefore a regular maintenance is needed to keep the data up-to-date. If the data repository is becoming large, this is only possible with a huge effort and as a result a lot of entries in a directory service might be outdated. In Venice, a service is automatically registered with the P2P network when it is deployed and active. Shutting down a service automatically results in de-registering it from the P2P network.

Therefore, using P2P technology allows to overcome these shortcomings and enhance the Information Broker with various beneficial properties:

- *Availability*. The use of a large number of peers allows for high availability of the meta data sets, because of data replication within the P2P network. While some of the peers are offline other peers can offer the requested meta data and therefore ensure availability.
- *Scalability*. Every peer of the network can be used to retrieve meta data and thus removes the performance bottleneck of a centralized registry.
- *Robustness*. While a centralized registry is a single point of failure, the distribution of meta data and the large number of access points to these meta data enhances the robustness of the Information Broker in contrast to centralized registries.

In order to lower the application requirements to use the Information Broker, the Information Broker provides a Web service interface to the client side. Therefore, the Information Broker is a dedicated service only concerned with data replication and retrieval, while providing transparent access to the P2P network for the requesting party, that does not directly participate in the P2P Network, but via a Web service interface [18]. Thus the clients requesting data from the P2P network do not have to use P2P software directly and don't have to open P2P ports on their operating system.

Service Information (SInfo). Every Venice service has to implement a special interface providing information about the service itself (name, purpose, icon, authorization scheme, etc.). Only by implementing this interface, the service can be seamlessly integrated into the framework.

2.3 (Horizontal) Application Services

Besides the management and informational services, several other useful services must be provided by a Service Grid in order to create a solid service foundation to build other services and applications upon. The horizontal application services within the Venice Service Grid can be used by any application domain and provide useful functionality of different kinds. Several services have already been developed and will be complemented by some more in the near future.

Property Storage Service (PSS). This service securely stores tag/value pairs. As simple as it might sound, its functionality has a large potential, e.g. for storing user preferences, service properties or other arbitrary data. This service is massively used by other services within the framework (e.g. the Notification Service or the Presence Service) in order to attain location transparency.

Time Keeper Service (TKS). This service can be used by other services or client programs to store measuring points consisting of a time stamp and a context. After measuring a sequence of data points, a time series analysis can be visualized and exported as image or text files. The results shown in section 3 have been measured and visualized using the Time Keeper Service. The overhead produced by the Time Keeper Service for each measurement is around 3500 ns.

Location Service (LOCS). When it comes to locating physical resources (printers, scanners, laboratory equipment, etc.) in three dimensional space, this service provides a geometric and symbolic positioning scheme and a useful plugin mechanism for integrating several location sensing techniques (like GPS, RADAR, etc) into clients. The Location Service provides standardized location information about humans or objects based upon the collected data of various different location systems. These location systems implement different location technologies, e.g. satellite based location technologies, cellular network based location technologies, indoor location technologies. Therefore all available data of the different location systems is used to compute the location of a person or object as exact as possible and to provide for the service user a transparent transition between different location systems. This finally allows for finding the nearest suitable resource for a specific task.

Notification Service (NS). A publish/subscribe notification service enables applications and services to subscribe for topics of interest and receive notifications when an event of interest appears. Such a service reduces polling and can even be used for deferred message delivery if a client or service is currently not running. Using the notification service it is possible to react to events occurring in the Service Grid by specifying

other services or tools that will be triggered by such an event. Every service implementing a special NotificationSink interface can receive notifications sent by the Notification Service. The services issuing notifications have to implement a NotificationSource interface so that it is possible to extract meta data about the potential notifications sent by the service.

Presence Service (PS). In order to strengthen user interaction inside the Venice Service Grid and to demonstrate the possibilities of the Information Broker, a Presence Service according to [19] is currently being developed. A presence service allows users to subscribe to each other and be notified of changes in state. Additionally, it is possible to apply certain rules of notification to groups of users. On top of the Presence Service and in combination with the Notification Service it will be possible to define actions that can be triggered when a certain state within a presence group appears.

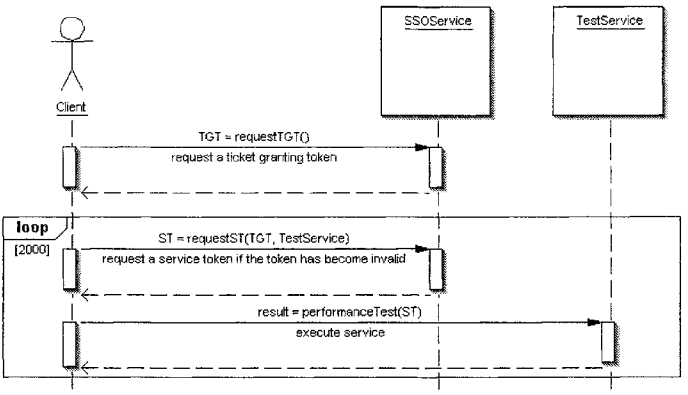


Figure 4. Performance Evaluation (UML Diagram)

2.4 (Vertical) Application Services

In a first application scenario, the framework has been used as an infrastructure for a Web services based Voice over IP (VoIP) application[1]. Several services dealing with VoIP functionality have been developed and brought into operation [7] (cf. *VoIP* in Fig. 2). A basic VoIP service abstracts from underlying VoIP protocols and gives access to both SIP and H.323 telephony. Additionally, phone calls can be made to the classic ISDN network by using an Asterisk server. Several additional supplementary services like Call Forwarding, Call Completion, a Voice Mailbox and a Call Center etc. make it worthwhile for the customers to use these VoIP services. The main goal was to move as much code as possible from the client to the services and to be able to seamlessly integrate new supplementary services or larger software updates without interruption.

[1] This research has been funded by Siemens AG, Munich.

In a current research project called DASIS[2] [20] (cf. *Virtual Office* in Fig. 2), the framework is used to virtualize the devices, tools and workflows of an ordinary office. Here, the Location Service will be responsible for locating physical resources like printers and scanners and associate it with other services or workflows.

3 Performance Evaluation

In order to evaluate the performance of the Venice Service Grid, the Time Keeper Service can be used to measure the time it takes to execute a specific task. The current time is calculated by a native (Windows or Linux) dynamic link library before and after a task. In this section the evaluation of an authorized Web service call as shown in Fig. 4 will be discussed. An evaluation of the SSO service itself has already been published in [12]. This measurement is not intended to compare the Venice Service Grid to other Grid technology; such a comparison would be unfair due to the different approaches of the diverse Grid solutions. It merely shows that the overhead produced by the SSO service (programmed in Java 2 5.0) is acceptable.

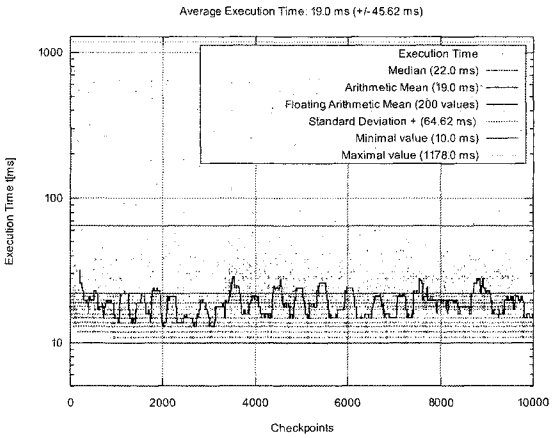


Figure 5. Performance Evaluation (Results)

The client has to request a TGT, request a ST and then subsequently call a simple Web service that checks the authorization of the client and returns the extracted username. This means, that at least some decryption of tokens has to be made by the service in order to extract the user name. Whenever the ST has become invalid, the client requests a new one from the SSO Service. In order to simulate a real service usage, the clients waits some time between the calls (randomly between 0 and 1 second).

[2] This research is part of the Cluster of Excellence "Dependable adaptive Systems and Mathematical Modeling" funded by the Program "Wissen schafft Zukunft" of the Ministry of Science, Education, Research and Culture of Rhineland-Palatinate, AZ.: 15212-52 309-2/40 (30)

In the evaluation scenario, five Java clients have been in operation simultaneously on five different hosts running under Windows and Linux. The Venice services have been deployed on a single Tomcat server (running under Linux with two Pentium 4 processors at 3 GHz). Server and clients have been connected by a 100 Mbit/s LAN. On the clients side, submitting the data to the Time Keeper Service adds an overhead of approx. 7 μs to each measurement.

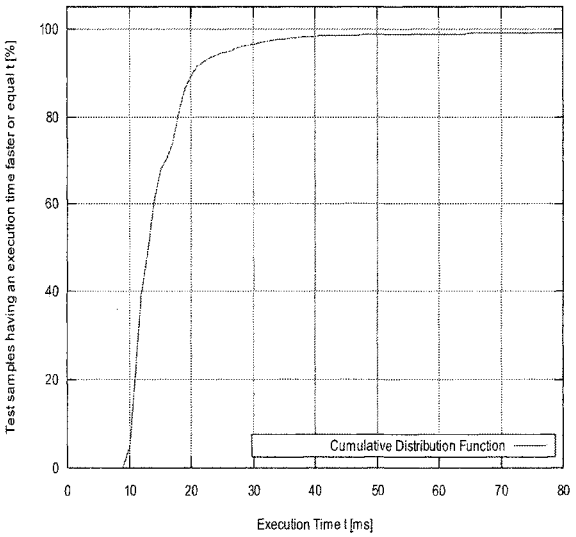


Figure 6. Performance Evaluation (Results)

Fig. 5 shows the time needed during 10,000 successful Web service calls. The arithmetic mean of a call is 19 ms. The standard deviation of the normally distributed value set is 64.62 ms. Additionally Fig. 6 shows the cumulative distribution of all test samples. Approximately 90 per cent of all request can be handled within 20 ms, 95 per cent within 25 ms and nearly all requests can be handled within 40 ms. This leads to the conclusion that there are only very few requests that consume a huge amount of time. A reason might be fluctuations of the network performance or within the Java virtual machine, e.g. garbage collection.

4 Conclusion and Future Work

This paper described the Venice Service Grid focussing on a lightweight approach for an easy to deploy, easy to maintain and easy to use Service Grid. The basic framework provides an open, dependable and flexible framework for deploying services to end-users. The architecture consists of Web services belonging to three different categories: Management Services, Information Services, and Application Services. The Management Services are necessary for the common usage and maintenance of Grid entities.

The Information Services collect, manage, and distribute information needed by users or services. As horizontal services the Application Services are typical services that can be used by any application that is build on top of this framework.

In the future the integration of the Web Services Resource Framework (WSRF) will allow for separating resources from services and – on the long run – allow for compatibility with other WSRF-enabled products like GT4. A connection to GT4, Unicore, gLite will transparently make available computing power and large data stores to all services and users inside Venice. The goal is to treat e.g. a GT4 node like a resource in Venice and thus enable all Venice services to acquire additional computing power or storage.

References

1. Foster, I.: What is the Grid? A Three Point Checklist. Grid Today, 2002
2. Foster, I., Kesselman, C. (Ed.): The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, 1998
3. UNICORE: http://unicore.sourceforge.net
4. gLite: Ligthweight Middleware for Grid Computing http://glite.web.cern.ch/glite/
5. Globus Toolkit 4. http://www.globus.org
6. The DataGrid project. http://eu-datagrid.web.cern.ch/eu-datagrid
7. Hillenbrand, M.; Götze, J.; Müller, J. and Müller, P.: Voice over IP – Considerations for a Next Generation Architecture. Proceedings of the 31th Euromico Conference (Porto, Portugal, 2005)
8. Apache Tomcat: http://tomcat.apache.org/
9. Apache Axis: http://ws.apache.org/axis/
10. De Clercq, J.: Single sign-on architectures Proceedings of Infrastructure Security, International Conference, InfraSec (Bristol, UK, 2002)
11. Opplinger R.: Authentication Systems for Secure Networks. Artech House (1996)
12. Hillenbrand, M., Götze, J., Müller, J. and Müller, P.: A Single Sign-On Framework for Web Services-based Distributed Applications. Proceedings of the 8th International Conference on Telecommunications ConTEL (Zagreb, Croatia, 2005)
13. Machiraju, V., Sahai, A., Moorsel A. v.: Web Services Management Network. Proceedings of the 8th IFIP/IEEE International Symposium on Integrated Network Management (Colorado Springs, US, 2003)
14. Zhang, G., Müller, J., Müller, P.: Designing Web Service Accounting Systems. Proceedings of the Workshop on Web Services: Business, Financial and Legal Aspects (Geneva, Switzerland, 2005)
15. Liebau N.-C., Darlagiannis, V., Mauthe, A., Steinmetz, R.: Token-based Accounting for P2P-Systems. Proceedings of Kommunikation in Verteilten Systemen KiVS (Kaiserslautern, Germany, 2005)
16. Hillenbrand, M., Müller, P. and Mihajloski, K.: A Software Deployment Service for Autonomous Computing Environments. Proceedings of the International Conference on Intelligent Agents, Web Technology and Internet Commerce IAWTIC (Gold Coast, Australia, 2004)
17. Forster, F., de Meer, H.: Discovery of Web Services with a P2P Network Proceedings of International Conference on Computational Science (2004)
18. Hillenbrand, M., Müller, P.: Web Services and Peer-to-Peer in: Peer-to-Peer Systems and Applications (Springer Verlag, Berlin, 2005)
19. Day, M., Rosenberg, J., Sugano H.: RFC 2778: A Model for Presence and Instant Messaging.
20. Project DASIS: http://www.dasmod.de

Semantic Integration of Identity Data Repositories

Christian Emig[1], Kim Langer[1], Jürgen Biermann[2], Sebastian Abeck[1]

[1] Cooperation & Management, Universität Karlsruhe (TH), 76128 Karlsruhe
[2] iC Consult GmbH, Keltenring 14, 82041 Oberhaching

{ emig | langer | abeck } @cm-tm.uka.de, biermann@ic-consult.de

Abstract. With the continuously growing number of distributed and heterogeneous IT systems there is the need for structured and efficient identity management (IdM) processes. This implies that new users are created once and then the information is distributed to all applicable software systems same as if changes on existing user objects occur. The central issue is that there is no generally accepted standard for handling this information distribution because each system has its own internal representation of this data. Our approach is to give a semantic definition of the digital user objects' attributes to ease the mapping process of an abstract user object to the concrete instantiation of each software system. Therefore we created an ontology to define the mapping of users' attributes as well as an architecture which enables the semantic integration of identity data repositories. Our solution has been tested in an implementation case study.

1 Introduction

The desire of enterprises to automate their business processes and to integrate existing IT solutions to enhance business performance spreads, among others, to the field of identity management (IdM). IdM can be defined as a set of processes and a supporting infrastructure for the creation, maintenance and use of digital identities (human users as well as IT systems) to enable efficient authentication, authorization and access control [1]. Processes in IdM include user provisioning, decommissioning and auditing [2]. To increase the automation of these processes, there is the need to provide an integrated view on the data which is being administered (cf. Figure 1), especially the user-specific data (i.e. the digital identities). This data is stored either standalone or can be directly attached to the business applications where it is employed to enforce access control. It is held in directories though other data storage solutions such as relational databases or XML-based files are conceivable as well. The repository may be distributed over different systems, and in case of a directory-based approach, the information in the repository is quite often accessible via the Lightweight Directory Access Protocol (LDAP). A digital identity is the representation of a subject that includes an identifier (e.g. a unique number), credentials and attributes. To enable efficient identity management it is important to keep the different identity repositories synchronized which implies an integration effort. The integration of heterogeneous data from different data repositories raises several questions that have not been fully answered yet.

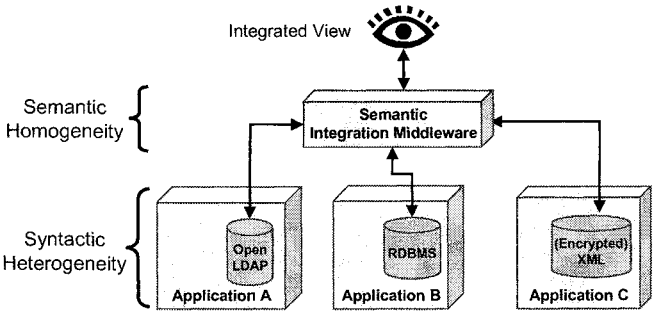


Figure 1. Semantic Integration of Heterogeneous Data

The integration of data can be performed on different tiers of data representation. The common integration approach is done on a syntactic level, merely pushing bytes from one integration end to the other with the meaning of data and so the mapping rules being hard-coded into the synchronization middleware. This approach though is rather shortsighted since the semantics of data are not being focused and changes in semantics entail direct changes within the synchronization middleware. Thus a more mature approach is needed which takes into account the semantics of the data objects processed. The term semantics in such a context is strongly correlated with ontologies. They play a key role in sharing a collective understanding of the semantics of the domain being described. Therefore, ontologies have to be considered when there is the need to elevate data repository integration to a semantic level and to provide an integrated view on data.

The two contributions of this paper are:

(1) The definition of an **extensible ontology** to enable semantic integration of syntactically heterogeneous identity data repositories (cf. chapter 3, "Person Ontology"). When addressing heterogeneity in this context it is important to point out that this heterogeneity is encountered on a syntactic and structural tier whereas the semantic tier appears quite homogenous, since all information on the user object in the domain of identity management is quite similar.

(2) An **architecture** supporting the semantic integration of identity data repositories (cf. chapter 4, "Data Repository Container"). Our approach wraps the application-specific user directories to a *data repository container* (DRC) by adding additional components to the integration architecture.

The paper is organized as follows: chapter 2 treats the related work about and the state-of-the-art of semantics in digital identities and how ontologies can ease the information integration in this context. Furthermore, architectural approaches are introduced that try to enhance directories to actively propagate local data changes to other directories. Chapter 3 and 4 describe our conceptual contributions which are applied in a case study illustrated in chapter 5. A conclusion and an outlook on future work in this area close the body of the paper.

2 Related Work and State-of-the-Art

2.1 Semantic Aspects

When dealing with the integration of different data sources it is for certain that data heterogeneity between the various data sources will be encountered. Problems referring to heterogeneity of data are already widely known within the distributed database systems community. In [3] it is distinguished between structural and syntactic heterogeneity (i.e. schematic heterogeneity) on the one hand and semantic heterogeneity (i.e. data heterogeneity) on the other hand. Structural heterogeneity means that different information systems store their data in different structures (e.g. schema). Semantic heterogeneity considers the contents of an information item and its intended meaning. In order to achieve semantic interoperability in a heterogeneous information system, the meaning of the information that is interchanged has to be understood across the systems. It is common knowledge that ensuring the semantic interoperability is much easier when staying in the same semantic domain [4, 5]. Though we are dealing with different IT systems that are not restricted to a specific domain, we can take advantage of the fact that it is not their business-related part that we are investigating. We look at very specific data objects: the users along with their identity and access management related properties.

There have been frequent discussions on how to handle heterogeneity when considering the semantic level. In [4], how to use ontologies in the context of information integration is discussed. The authors suggest reducing the hard-coding which does the translation between the terminologies of pairs of systems by applying ontologies to the formal specification of the meaning of terms. According to [6], three different directions can be identified when applying ontologies: single ontology approaches, multiple ontologies approaches and hybrid approaches being a mixture of both. Single ontology approaches use one global ontology providing a shared vocabulary for the specification of the semantics. All information sources are related to this global ontology. Single ontology approaches can be applied to integration problems where all information sources to be integrated provide nearly the same view of a domain. Using multiple ontology approaches, each information source is described by its own ontology. Quite often this moves the complexity to an intermediary ontology which is needed to enable the matching between the different domains. Hybrid approaches work with local ontologies as well but try to use a globally shared vocabulary or glossary, which is to be used by the human developer when designing the local ontologies. For our approach of enabling semantic integration of identity data repositories we can take advantage of the fact that the semantics of user-centric identity data have a common core which can be instantiated at almost all of the existing IT systems. This facilitates the creation of a single ontology approach describing a person and his/her attributes which can then be mapped to the different IT systems.

The main causes for semantic heterogeneity are classified in [7]. In our case the central focus at the semantic level is to address so-called "naming conflicts" that occur when naming schemes of information differ. "Scaling conflicts" and "confounding conflicts" are not to be expected in our scenario. Naming conflicts can oc-

cur if the attribute names of digital identities differ at the various systems. For example, think of *givenName='John'* and *firstName='John'*. This problem can be addressed by introducing synonyms which can be handled using ontologies. The granularity of information is a further point to be solved. Problems occur if in the one repository the attributes *firstName* and *lastName* exist whilst in the other repository only the concatenation of both, called *commonName* can be found. This can also be solved using ontologies.

The application of ontologies in integration scenarios is described in [5], which concentrates at the process for the definition of ontologies. The ontology design process is regarded as a bottom-up approach taking the schemas of multiple databases as input and producing as output a single unified database schema combined with a mapping from the individual databases to the unified database. As the number of IT systems that apply identity management and access control and therefore need user objects is enormous, a starting point has to be found where the parts of users' data are defined which are not application-specific. This is where the state-of-the-art concerning existing proposals about how to describe users has to be investigated.

Most of the present suggestions concerning how to describe a user are rather simple – they only deal with the basic properties a person can have. As a starting point the nearest idea is to examine the directories where the users' digital identity information is stored. The leading standard in this context is LDAP [8]. Though being expandable by creating new schemata or schema extensions the focus is clearly on the syntactic and structural layer. Attributes have "meaningful names" and can be described using plain text. For the integration of user data objects over more than one LDAP-based directory, the schema has to be interchanged and implemented at all participating directories, which reduces flexibility in the integration process. To set up on the syntactically defined data, simple ontologies have been defined [9, 10]. Further ontologies such as the "Enterprise Ontology" developed by the University of Edinburgh (available at the ontology library at [11]) also include rather primitive person classes. There are some other suggestions such as the HR-XML initiative [12]. The HR-XML consortium is a non-commercial and independent organization responsible for the release of a standard human resource vocabulary. Though no knowledge representation language such as OWL (Ontology Web Language, [13]) is used to describe the included items, we do use it as a starting point to set-up our *person ontology* as described in chapter 3.

2.2 Architectural Aspects

It is not enough to address only semantics when looking at identity data repository integration. There must be a way to handle the actual synchronization process over the different repositories. As repositories usually act passively with respect to event propagation, it must be possible to determine if a change has occurred in the repository. Architecturally different approaches are available which solve this problem. There is the retro change log (RCL) [14] featured by Sun which is implemented as an additional sub-tree in the directory server. Using RCL, a record of each change made to the directory server is stored by duplicating changed objects to a specific sub-tree. The idea is that external synchronization software checks this sub-tree, takes the in-

formation and finally deletes the entries there. Due to the tight alignment with the LDAP basis standards this is a very interoperable approach. A major deficiency is the need to regularly poll the directory there is nothing like a publish/subscribe-based event propagation. Problems occur if there is more than one remote directory trying to synchronize, because it is not defined when the last party is informed about the changed object and when the object can be removed.

The need for a proactive notification in case of changes inside the repository is described in [15]. There it is argued that an exclusively inactive (i.e. passive) interface to directory services can hinder server scalability and indirectly restrict the behavior of potential applications. So the authors propose to extend directory services' interfaces with a proactive mode with which clients can express their interest in changes in the environment according to a publish/subscribe paradigm. The problem of this approach is that only the passiveness is overcome but the semantic heterogeneity is not addressed which is still to be handled individually on a 1:1 basis by the synchronization application. Another solution for enhancing data repositories to proactively propagate data changes is described in [16] but again not on a semantic basis. The authors implemented an effect similar to a trigger in databases by adding an intercepting gateway which is capable of starting the propagation of data to different directories based on the content of the invoked action in front of their LDAP-based directory. This approach moves directories out of their role as a passive data source but it is strongly bound to LDAP-based directories. There is nothing like an aid in semantic matching as the point-to-point specific rules have to be hard-coded at the gateway.

3 Person Ontology

An ontology helps to separate the meaning of data from its representation. Thus the meaning of data is extracted from applications, databases and directories, and can be altered independently without having to adjust the data's representation itself. This approach offers different advantages. First of all, the use of ontologies can provide a unified nomenclature for the entities of the domain of interest. It also yields semantic uniqueness, which implies that entities in the ontology have a distinguishable and semantically well-defined meaning. Therefore it can be prevented that items that are syntactically (e.g. their names) but not semantically equal are believed to have the same meaning. Furthermore the use of an ontology leads to a more flexible integration since hard-coding of the meaning of data is prevented and changes to that meaning can be made without having to cope with the data itself. Another benefit of ontologies is that once a conceptualization of the specific domain has been accomplished, it is possible to share this knowledge in a well-defined and formal manner so that other parties are able to use that knowledge directly. Moreover, if an ontology is at hand it is possible to conduct consistency checks on its extension. Beyond that it is feasible to extract implicit knowledge from the ontology's extension. These facts lead to the conclusion that sophisticated identity management should be accomplished using ontologies. In the following an ontology representing the meaning of the person object will be elaborated in order to overcome the syntactic heterogeneity of user data within different data repositories.

The development of the *person ontology* has been tightly aligned to the approach described in [17]. The first issue that has been addressed in this context is the description of the ontology's domain. Basically that is the domain of IdM or more precisely persons' or users' identity data. After defining the domain, the next step is to investigate how the ontology will be used and what audience it appeals to. The answer to the first of these concerns is that it is designed for the integration of syntactically heterogeneous data sources holding digital identities. This implies that the main audience to make use of and augment the ontology will be essentially system integrators and identity managers responsible for the establishment of processes such as provisioning, synchronization and decommissioning.

With these prerequisites made explicit we started to enumerate the most important attributes associated with persons. The heuristics to do so is considering the importance of the attributes, or in other words the frequency they are encountered, from existing data representations such as LDAP or HR-XML. Some of the elements that we have distinguished this way are *commonName*, *address*, *credentials*, *title*, *email*, *telephone, fax* and *birthday*. As the next step we split up the elements into (simple) datatype properties (i.e. attributes) and into complex elements, which we defined as classes and put into a hierarchy using OWL. Examples for complex types are *commonName, address* and *credentials*. Additionally, the relationships between these classes were formalized using object properties. Thereby classes, such as *commonName* and *cn* have been characterised to be semantically equivalent. However classes that have divergent meanings such as *userPassword* from LDAP and *password* from HR-XML were marked to be semantically different. Finally the primitive attributes of each class had to be defined. This was done by using the datatype property construct provided by OWL. Thereby a special focus was set on the definition of semantically alike properties to automate the mapping process. To give a short example of the possible connections between classes, a part of the person class is described: A person has the *hasAddress* object properties that associates him with the *address* class. This *address* class in turn is a super-class to further classes such as *postalAddress*. A *postalAddress* again is connected to a *commonName* via the *hasCN* object property and holds datatype properties such as *street, zipCode, country*. A *street* is assembled from its *streetName* along with a *houseNumber* with and an optional *suffix*. This is just a small excerpt from the complete ontology. The major part of the ontology is reserved for the definition of syntactically divergent terms that have semantically the same meaning such as the attributes *mobile*, *mobilePhone* and *mobilePhoneNumber*.

We have developed the *person ontology* using Protégé [18] Version 3.1. The tool we used to verify OWL-DL conformity is the OWL Validator of WonderWeb [19]. To assure the correctness and consistency of the *person ontology* we deployed the Racer DIG (Description Logic Implementation Group) Reasoner [20]. In Figure 2 we depict an extract of our *person ontology*, graphically modelled using DLG² [21]. DLG² is a graphically-based language that can be used to simplify the presentation of RDF (Resource Description Framework) and therefore OWL models. The idea is to exemplary show the relevant constructs that we have applied in the ontology in a human readable manner. DLG² enables for a flexible modelling of datatype properties. They can either be defined inside a class or externally in an ellipse to allow modelling of equivalent properties.

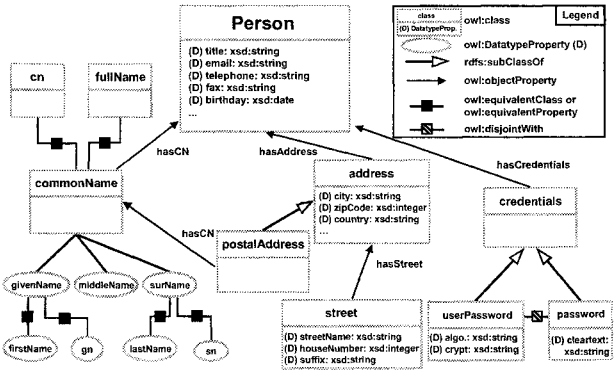


Figure 2. Person Ontology (Extract in DLG²)

4 Data Repository Container

Defining the semantics of users' digital identities using a global ontology builds the first element of the integration's core. The second is a suitable architecture building on this ontology and enabling the repository integration as traditional data sources are not capable of this. At a glance our approach is to extend the traditional, passive identity data sources by adding further components. We call an enhanced data source *data repository container* (DRC). Though a DRC still can act locally and autonomously, it is to be attached to a specific kind of message broker to which each DRC subscribes in order to both publish and receive information on changed user objects. In the following sub-chapters we introduce the architecture that we have developed.

4.1 Functional Requirements and Approach

The following functional requirements led to the development of the *data repository container* (DRC):

1. The DRC should be capable of handling any kind of common data storage technology, such as directories, relational database management systems (RDBMS), XML files or any similar kind of technology. This should be achieved by introducing an abstraction layer which we called *data source wrapper* (DSW).

2. To be able to align an incoming user object with the representation of the local repository, we employ a *semantic engine* (SE) which does the mapping and filtering of the incoming attributes to the local ones with the help of the *person ontology*.

3. Changes in the local source repository must be propagated proactively by the DRC instead of making the destinations poll the source on a regularly basis. As a means of interaction, a message-oriented approach should be pursued to enable a flexible, reliable and loosely-coupled mechanism for the information interchange. This functionality is put into a component that we call *semantic event dispatcher* (SED) that is located inside each DRC and takes care of both sending out and receiving these event messages containing the changed user objects.

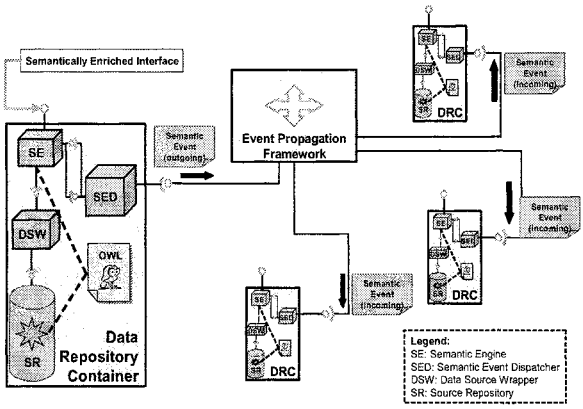


Figure 3. Architecture for Semantic Directory Integration

4. With a growing number of DRCs, a point-to-point connection between all the containers results in high integration costs due to the n-squared problem when new *data repository containers* are being added. This can be avoided by setting up a message broker which we call *event propagation framework* (EPF) and to which every DRC is connected. Unlike traditional solutions, the basic setup of this EPF can be rather simple. By offering a standardized interface it is fed by its corresponding SED if a user object is created or changed (outgoing semantic event from the view of a DRC) and distributes this information to all DRCs participating (incoming semantic event).

4.2 Data Source Wrapper

The *data source wrapper* (DSW) acts as a converter between the underlying technology dependent protocol, e.g. LDAP, and a common protocol used to access it. Its design is based on the wrapper and adapter design pattern described in [22]. The fundamental idea behind a wrapper or adapter is to map the interface of a class, in this case the interface of the source repository, to an interface expected by a client. Thus it solves the problem of interoperability between incompatible interfaces. This is especially important to reuse functionality; in our case it enables the reuse of the components SE and SED. Compared with the SE that is described in the next section and which accomplishes semantic data integration, the DSW's job is to ensure interoperability between the SE and the elements on the data layer. Thus it accomplishes a syntactic integration of the source repositories. In conclusion, it becomes possible to disengage from the underlying protocol and present a uniform interface to the SE.

4.3 Semantic Engine

The *semantic engine* (SE) constitutes the central element within the DRC's architecture. Its responsibility is to semantically integrate person data from different sources based on the *person ontology*. Thus it performs the mapping of users' properties. This means that the SE basically does a semantic transformation from an incoming person object, either coming from its regular service interface or via notification from other

DRCs via the EPF, to a person object that is expected by the underlying, local repository. The SE is based on the proxy design pattern as described by [22]. Therefore it functions as an interface to the data repository. Accordingly all interaction with the data repository has to pass the SE. This enables the SE to detect all relevant changes in the repository. Moreover, the SE can raise events based on the actions performed on the repository and dispatch these events via the *semantic event dispatcher* (SED).

4.4 Semantic Event Dispatcher

The *data repository containers* should be able to exchange events in a flexible, reliable and standardized way. To uphold the principle of separation of concerns the SE should not be responsible for communication issues. Therefore another component is needed that hides the complexity of message exchange to the SE. This is precisely the task of the *semantic event dispatcher* (SED). Its purpose is to expose an interface to the SE, thus making the interaction logic transparent for the SE. If an event has been detected by the SE, it is passed to the SED which takes care of the communication with the *event propagation framework*. The SED is designed according to the design pattern façade as introduced by [22]. All logic involving the distribution of events is delegated to the SED. Events are published on so-called topics. A conceivable example of topics in such a context could be <hostname>.update or <hostname>.create. To get the full information, the subscription of *.* is recommended, but DRCs that are interested only in updates from a specific DRC could subscribe to the topic <hostname>.* at the EPF. Parallel to the distribution of events, the SED is also accountable for the reception of events by subscribing to the relevant topics at the EPF.

4.5 Event Propagation Framework

A central *event propagation framework* (EPF) is a connector between the DRCs that reduces point-to-point communication between different DRCs. It acts as the authority responsible for the distribution of the changes in user objects. The DRCs subscribe at the EPF and there is the possibility to define different topics. The tagging of the events to specific topics is done by the sending SED, so the EPF acts as a message broker with all the features as asynchronicity and loose-coupling.

4.6 Collaboration Aspects

Figure 4 illustrates how the components forming the *data repository container* work together. This is exemplary shown by a DRC receiving an (incoming) event from the EPF.

An event sent by the EPF is received by the SED. Each event is associated with a certain topic it is published on. In order to receive events the SED must have previously subscribed to the appropriate messaging topic at the EPF. After the SED has received an event it is passed on to the SE using the *processEvent* method. The SE takes on the semantic processing of the event by aligning its syntax to the syntax of the local DSW based on the *person ontology*. After this has been accomplished the appropriate action on the source repository is executed. In this case this action is a *delete* operation.

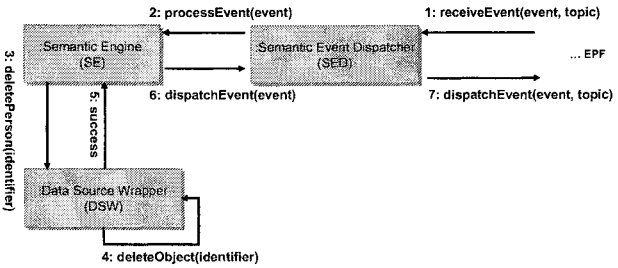


Figure 4. Data Repository Container – Collaboration Diagram

After the action has successfully been performed by the DSW the SE dispatches a new event since a deletion has occurred. This message is again passed to the SED. The SED in turn posts the event on a certain topic (e.g. <host-name>.acknowledgement) so that further DRCs are accurately notified and data consistency is assured. This enables other DRCs or a centralized auditing system to check if all DRCs have processed a specific event.

5 Implementation Case Study

We have developed our solution to fit into a project context at a major automotive company. Though our design is mostly technology independent, the preference was to use Java technology for the actual implementation. With the blue-print of a future service-oriented architecture (SOA) that is planned, the outer interface of the DRC was to be implemented as a web service which implies a WSDL-style interface description as well as SOAP communication. JBoss was the choice considering the necessary application server which a DRC is deployed to. The wrapping to web service interfaces is done by the JBoss internal component WS4EE. The three components SE, SED and DSW are implemented as Enterprise Java Beans (EJB). The *semantic engine* is a stateless session bean and its interface is exposed as a web service as an enhancement to the simple and proprietary interface of the source repository. The *semantic event dispatcher* is implemented as a message driven bean and utilizes the concepts of Java Messaging Services (JMS) to communicate with the EPF and the SE. The *data source wrapper* is implemented as an entity bean representing a standardized storage for the user objects. The components and their coupling are depicted in the architectural overview in Figure 5. The *semantic engine* is supported by a helper class, the *semantic mapper* class. This class does the semantic mapping of and creation of person objects based on the *person ontology* elaborated in chapter 3. To access the ontology we use the JENA API [23]. Mapping and transformation are done based on the knowledge given by the ontology. This means that if further mappings have to be defined only the ontology must be altered but not the code of the EJBs.

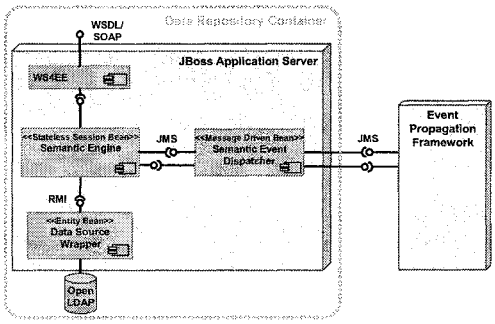


Figure 5. Case Study – Implementation of a Data Repository Container

In this scenario, various directories had to be synchronized. There was a distinction between offline and online synchronization. The latter means that change events are propagated short time after they occurred. We focused on the online synchronization. The synchronization application in use was proprietarily developed with individual logic for point-to-point synchronization. We set up the EPF as a central message broker and attached the corporate meta directory (Sun Directory Server) as well as the Microsoft Active Directory, PeopleSoft ePeople and Lotus Notes. The overall amount of classes that have been defined in the ontology is 9. It further contains 71 datatype and object properties. The OWL file has 800 lines and is about 40 kilobytes in size.

6 Conclusion and Further Work

In this paper a solution for the semantic integration of person objects has been presented. We have introduced a core set of a *person ontology* that can be flexibly extended as well as an architecture enhancing traditional identity repositories to active and semantic-enabled *data repository containers*. These enable the integration process for user provisioning, decommissioning and synchronization. For the loose coupling of the different *data repository containers* we developed the *event propagation framework* as a message broker. It allows the distribution of events between the different DRCs and centralizes the point-to-point connections.

Currently the development of the ability to dispatch events without the need of an active change in the source directory is a main issue which must be solved. For example the date of a person leaving the company is quite often recorded in advance, so the SED should be able to dispatch the event automatically at the point of time it is needed for the overall decommissioning process. Another issue is the implementation of natural language processing for the *semantic engine* in order to allow a declarative data source access. In addition, the main focus of the *person ontology* currently is to ease the mapping of attributes' names. We would like to enhance our approach to address the mapping of the attributes' values as well (e.g. role mapping). With upcoming service-oriented architecture (SOA) in mind, we have already applied SOA paradigms like loose coupling and web service interfaces achieving better interoperability. For a further SOA alignment, the embedding of the *event propagation framework* to the enterprise service bus (ESB) of an SOA is to be tightened.

7 References

1. Burton Group: Concepts and Definitions (Glossary), Version 2.0, September 2005.
2. Phillip J Windley: Digital Identity, O'Reilly Media; 1st edition, August 2005.
3. V. Kashyap and A. Sheth: Schematic and semantic semilarities between database objects: A context-based approach. The International Journal on Very Large Data Bases, 5(4):276–304, 1996.
4. Zhan Cui, Dean Jones and Paul O'Brien: Issues in Ontology-based Information Integration, IJCAI Seattle / USA, 2002.
5. Chris Partridge: The Role of Ontology in Semantic Integration, OOPSLA 2002, Seattle.
6. H.Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner: Ontology-Based Integration of Information - A Survey of Existing Approaches, Intelligent Systems Group, Center for Computing Technologies, University of Bremen, 2001.
7. Cheng Hian Goh. Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources, MIT, 1997.
8. Lightweight Directory Access Protocol (v3). URL: http://www.ietf.org/rfc/rfc2251.txt
9. Li Ding, Harry Chen, Lalana Kagal, Tim Finin: DAML Person Ontology, 2002. URL: http://daml.umbc.edu/ontologies/ittalks/person
10. UMBC Ebiquity Research Group: Person Ontology. URL: http://ebiquity.umbc.edu/ontology/person.owl
11. Standord University: Knowledge Systems Laboratory Ontology Editor, June 2006. URL: http://www-ksl-svc.stanford.edu:5915/
12. Homepage of the HR-XML Consortium. URL: http://www.hr-xml.org/
13. World Wide Web Consortium (W3C): OWL Web Ontology Language Overview, W3C Recommendation, February 2004. URL: http://www.w3.org/TR/owl-features/
14. Sun: Retro Change Log Plug-In. URL: http://docs.sun.com/source/816-6698-10/replicat.html#15790
15. Fabian E. Bustamante, Patrick Widener and Karsten Schwan: A Case for Proactivity in Directory Services, Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (HPDC), 2002.
16. Robert Arlein, Juliana Freire, Narain Gehani, Daniel Lieuwen, and Joann Ordille: Making LDAP Active with the LTAP Gateway, in Proceedings Workshop on Databases in Telecommunication, September 1999.
17. Natalya F. Noy and Deborah L. McGuinness: Ontology Development 101: A Guide to Creating Your First Ontology, Stanford University, Stanford, CA, 94305, 2001.
18. Stanford Medical Informatics: Protégé Ontology Editor, 2005. URL: http://protege.stanford.edu
19. Sean Bechhofer and Raphael Volz: WonderWeb OWL Ontology Validator, 2003. URL: http://phoebus.cs.man.ac.uk:9999/OWL/Validator
20. Racer DIG Reasoner, July 2006. URL: http://www.racer-systems.com/de/index.phtml, http://dig.sourceforge.net/
21. Xiaoshu Wang, Jonas S. Almeida: DLG2 - A Graphical Presentation Language for RDF and OWL, 2005. URL: http://charlestoncore.musc.edu/docs/dlg2.html
22. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns, Addison Wesley, 1998.
23. Hewlett-Packard Development Company: JENA – A Semantic Web Framework for Java, 2005. URL: http://jena.sourceforge.net

Throughput Performance of the ActiveMQ JMS Server

Robert Henjes, Daniel Schlosser, Michael Menth, and Valentin Himmler

University of Würzburg, Institute of Computer Science
Am Hubland, D-97074 Würzburg, Germany
Phone: (+49) 931-888 6644, Fax: (+49) 931-888 6632
{henjes,schlosser,menth,himmler}@informatik.uni-wuerzburg.de

Abstract. Communication among distributed software components according to the publish/subscribe principle is facilitated by the Java messaging service (JMS). JMS can be used as a message routing platform if the subscribers install filter rules on the JMS server. However, it is not clear whether its message throughput is sufficient to support large-scale systems. In this paper, we investigate the capacity of the high performance JMS server implementation ActiveMQ. In contrast to other studies, we focus on the message throughput in the presence of filters and show that filtering reduces the performance significantly. We present a model for the message processing time at the server and validate it by measurements. This model takes the number of installed filters and the replication grade of the messages into account and predicts the overall message throughput for specific application scenarios.

1 Introduction

The Java Messaging Service (JMS) is a communication middleware for distributed software components. It is an elegant solution to make large software projects feasible and future-proof by a unified communication interface which is defined by the JMS API provided by Sun Microsystems [1]. A salient feature of JMS is that applications can communicate with each other without knowing their communication partners as long as they agree on a uniform message format. Information providers publish messages to the JMS server and information consumers subscribe to certain message types at the JMS server to receive a certain subset of these messages. This is known as the publish/subscribe principle.

In the non-durable and persistent mode, JMS servers efficiently deliver messages reliably to subscribers that are presently online. Therefore, they are suitable as backbone solution for large-scale realtime communication between loosely coupled software components. For example, some user devices may provide presence information to the JMS. Other users can subscribe to certain message types, e.g. the presence information of their friends' devices. For such a scenario, a high performance routing platform needs filter capabilities and a high capacity to be scalable to a large number of users. In particular, the throughput capacity of the JMS server should not suffer from a large number of clients or filters.

In this paper we investigate the performance of the ActiveMQ [2] JMS server implementation. We evaluate the maximum throughput by measurement under various conditions. In particular, we consider different numbers of publishers, subscribers, and

This work was funded by Siemens AG, Munich. The authors alone are responsible for the content of the paper.

filters, different kinds of filters, and filters of different complexity to characterize the throughput performance of the ActiveMQ JMS Server. We also propose a mathematical model depending on the number of filters and the message replication grade to approximate the processing time of a message for the ActiveMQ server.

The paper is organized as follows. In Section 2 we present JMS basics that are important for our study and consider related work. In Section 3 we explain our test environment and measurement methodology. Section 4 shows measurement results and proposes a model for the processing time of a simple message depending on the server configuration. These models are useful to predict the server throughput for specific application scenarios. Finally, we summarize our work in Section 5.

2 Background

In this section we describe the Java messaging service (JMS) and discuss related work.

2.1 The Java Messaging Service

Messaging facilitates the communication between remote software components. The Java Messaging Service (JMS) is one possible standard of this message exchange. So-called publishers connect to the JMS server and send messages to it. So-called subscribers connect to the JMS server and consume available messages or a subset thereof. So the JMS server acts as a relay node [3], which controls the message flow by various message filtering options. This is depicted in Figure 1. Publishers and subscribers rely on the JMS API [1] and the JMS server decouples them by acting as a broker. As a consequence, publishers and subscribers do not need to know each other.

The JMS offers two different connection modes: a durable and a non-durable connection type. If a subscriber connects in the durable mode, the messages will be stored for delivery if this client disconnects. All stored messages will be delivered when the client connects next time to the JMS server. Persistence is another option for JMS. If the persistent option is set, each message has to be delivered reliably to all actively connected clients, which is ensured by confirming reception with acknowledgments. In the non-persistent mode the JMS server must deliver the message only with an at-most-once guarantee. This means that the message can be lost, but it must not be delivered twice according to [1]. In this study, we only consider the persistent but non-durable mode.

Information providers with similar themes may be grouped together by making them publish to a so-called common „topic"; only those subscribers having subscribed for that specific topic receive their messages. Thus, topics virtually separate the JMS server into several logical sub-servers. Topics provide only a very coarse and static method for message selection due to the fact that publishers and subscribers have to know which topics they need to connect to. This results in a slight loose of the decoupling feature in the publish/subscribe context. In addition, topics need to be configured on the JMS server before they can be used actively. If no topics are explicitly introduced at the JMS server, exactly one default topic is present, to which all subscribers and publishers are connected.

Filters are another option for message selection. A subscriber may install a message filter on the JMS server. Only the messages matching the filter rules are forwarded to the

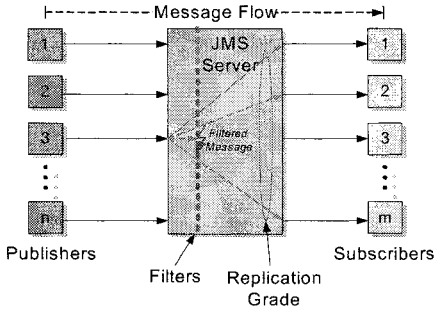


Fig. 1. The JMS server delivers messages from the publishers to all subscribers with matching filters.

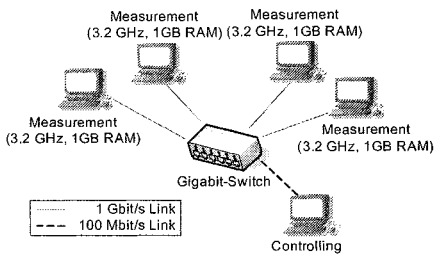


Fig. 2. Testbed environment.

respective subscriber instead of all messages. In contrast to topics, filters are installed dynamically during the operation of the server by each subscriber.

A JMS message consists of three parts: the message header, a user defined property header section, and the message payload itself [1]. So-called correlation IDs are ordinary 128 byte strings that can be set in the fixed header of JMS messages as the only user definable option within this header section. Correlation ID filters try to match these IDs whereby wildcard filtering is possible, e.g., in the form of ranges like [#7;#13], which means all IDs between #7 and #13 are matched including #7 and #13. Several application-specific properties may be set in the property section of the JMS message. Application property filters try to match these properties. Unlike correlation ID filters, a combination of different properties may be specified which leads to more complex filters with a finer granularity. After all, topics, correlation ID filtering, and application property filtering are three different possibilities for message selection with different semantic granularity and they require different computational effort.

2.2 Related Work

The JMS is a wide-spread and frequently used middleware technology. Therefore, its throughput performance is of general interest. Several papers address this aspect already but from a different viewpoint and in different depth.

The throughput performance of four different JMS servers is compared in [4]: FioranoMQ [5], SonicMQ [6], TibcoEMS [7], and WebsphereMQ [8]. The study focuses on several message modes, e.g., durable, persistent, etc., but it does not consider filtering, which is the main objective in our work. The authors of [9] conduct a benchmark comparison for the SunMQ [10] and IBM WebsphereMQ. They tested throughput performance in various message modes and, in particular, with different acknowledgement options for the persistent message mode. They also examined simple filters, but they did not conduct parametric studies, and no performance model was developed. The objective of our work is the development of such a performance model to forecast the maximum message throughput for given application scenarios. A proposal for designing a "Benchmark Suite for Distributed Publish/Subscribe Systems" is presented in [11] but without measurement results. The setup of our experiments is in line with these recommendations. General benchmark guidelines were suggested in [12] which apply both

to JMS systems and databases. However, scalability issues are not considered, which is the intention of our work. A mathematical model for a general publish-subscribe scenario in the durable mode with focus on message diffusion without filters is presented in [13] but without validation by measurements. The authors of [13] present in [14] an enhanced framework to analyze and simulate a publish/subscribe system. In this work also filters are modeled as a general function of time but not analyzed in detail. The validation of the analytical results is done by comparing them to a simulation. In contrast, our work presents a mathematical model for the throughput performance in the non-durable mode including several filter types and our model is validated by measurements on an existing implementation of a JMS server. Several other studies address implementation aspects of filters. A JMS server checks for each message whether some of its filters match. If some of the filters are identical or similar, intelligent optimizations may be applied to reduce the filter overhead [15].

The Apache working group provides the generic test tool JMeter for throughput tests of the ActiveMQ [16]. However, it has only limited functionality such that we rely on an own implementation to automate our experiments.

3 Test Environment

Our objective is the assessment of the message throughput of the ActiveMQ JMS server by message under various conditions. For comparability and reproducibility reasons we describe our testbed, the server installations, and our measurement methodology in detail.

3.1 Testbed

Our test environment consists of five computers that are illustrated in Figure 2. Four of them are production machines and one is used for control purposes, e.g., controlling jobs like setting up test scenarios and monitoring measurement runs. The four production machines have a 1 Gbit/s network interface which is connected to one exclusive Gigabit switch. They are equipped with 3.2 GHz single CPUs and 2048 MB system memory. Their operating system is SuSe Linux 9.1 with kernel version 2.6.5-smp installed in standard configuration. The "smp"-option enables the support of the hyper-threading feature of the CPUs. Hyper-threading means that a single-core-CPU uses multiple program and register counters to virtually emulate a multi-processor system. In our case we have two virtual cores. To run the JMS environment we installed Java JRE 1.5.0 [17], also in default configuration. The control machine is connected over a 100 Mbit/s interface to the Gigabit switch. In our experiments one machine is used as a dedicated JMS server. Our test application is designed such that JMS subscribers or publishers can run as Java threads. Each thread has an exclusive connection to the JMS server component and represent a so-called JMS session. A management thread collects the measured values from each thread and appends these data to a log file in periodic intervals.

In our test environment the publishers run on one or two exclusive publisher machines, and the subscribers run on one or two exclusive subscriber machines depending on the experiment. If two publisher or subscriber machines are used, the publisher or subscriber threads are distributed equally between them.

3.2 Server Installation

The ActiveMQ server version 4.0 stable [2] is an open source software provided by the Apache group. We installed it on one of the above described Linux machines in default configuration such that the hyper-threading feature of the Linux kernel is used and the internal flow control is activated. To ensure that the ActiveMQ JMS server has enough buffer memory to store received messages and filters we set explicitly the memory for the Java Runtime Environment to 1024 MB.

3.3 Measurement Methodology

Our objective is the measurement of the JMS server capacity and we use the overall message throughput of the JMS server machine as performance indicator. We keep the server in all our experiments as close as possible to 100% CPU load. We verify that no other resources on the server machine like system memory or network capacity are bottlenecks. The publisher and subscriber machines must not be bottlenecks. Therefore their CPU loads must be lower than 75%. To monitor these side conditions, we use the information provided in the Linux „/proc" path. We monitor the CPU utilization, I/O, memory, and network utilization for each measurement run. Without a running server software, the CPU utilization of the JMS server machine does not exceed 2%, and a fully loaded server must have a CPU utilization of at least 95%.

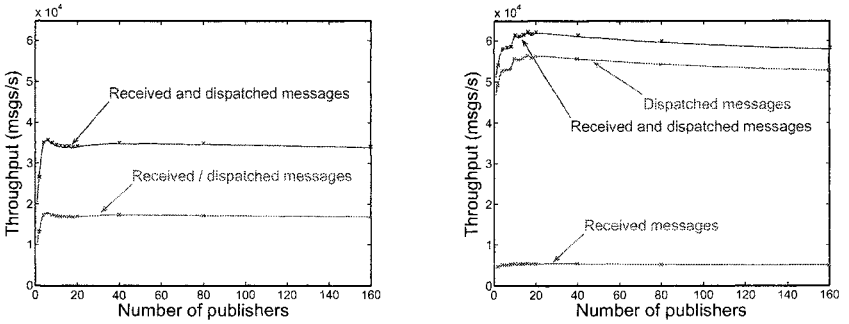
Experiments are conducted as follows. The publishers run in a saturated mode, i.e., they send messages as fast as possible to the JMS server. However, they are slowed down by the flow control of the server such that we observe publisher-side message queueing. We count the overall number of sent messages at the publishers and the overall number of received messages by the subscribers to calculate the server's rate of received and dispatched messages. Our measurement runs take 10 minutes whereby we discard the first seconds due to warmup effects. For verification purposes we repeat the measurements several times, but their results hardly differ such that confidence intervals are very narrow even for a few runs. Therefore, we omit them in the figures of the following sections. The following experiments use the non-durable and persistent messaging mode as described in the Section 2.

4 Measurement Results

In this section we investigate the maximum message throughput of the ActiveMQ JMS server. The objective is to assess and characterize the impact of specific application scenarios on the performance. In particular, we consider filters since they are essential for the use of a JMS server as a general message routing platform.

4.1 Impact of the Number of Publishers and Subscribers

In our first experiment, we study the impact of different numbers of publishers and subscribers on the message throughput. The results of the following two experiments yield the minimum number of clients which have to be connected to the JMS server to fully load the JMS server. In the persistent mode, i.e., lost messages are retransmitted by the JMS server and messages are preliminarily written to a disk for recovery purposes. Also each message is explicitly acknowledged by the recipient of the message.



(a) 1 subscriber thread ($r = 1$) (b) 10 subscriber threads ($r = 10$)

Fig. 3. Impact of the number of publishers on the message throughput

Impact of Number of Publishers We study the impact of the number of publishers for a single connected subscriber without filters and for 10 connected subscribers without filters. We present the throughput of the messages that are received and dispatched by the server in Figure 3(a) and Figure 3(b) together with their sum which we call the overall message throughput in the following.

For a single subscriber, the throughput of received messages equals the one of dispatched messages since each message is forwarded to only one subscriber while for 10 subscribers, the dispatched throughput is 10 times larger than the received throughput. As the overall throughput of the server is limited, the received throughput for 10 subscribers is clearly smaller than the one for a single subscriber. Thus, the average number of replications of each message clearly impacts the received throughput and we call it the replication grade r in the following. A comparison of the absolute throughput of both experiments shows that only a relatively small overall throughput of 34000 msgs/s can be achieved for a single subscriber while for 10 subscribers, a maximum overall throughput of 62000 msgs/s can be achieved for 16 publishers.

For both experiments, the throughput is almost independent of the number of publishers if this number is sufficiently large (about 10) such that we use at least 10 subscribers in the following experiments. We also observed that the server cannot be fully loaded with a single subscriber.

Impact of the Number of Subscribers Similarly to the experiment above, we investigate the impact of the number of subscribers on the JMS server throughput. To that end, we have 20 publisher threads running on one machine and vary the number of subscribers on two other machines. Figure 4 shows the received, dispatched, and the overall message throughput. The maximum overall throughput of about 60000 msgs/s is reached for 10 connected subscribers and decreases decreases only slightly for an increased number of subscribers. The received throughput of the JMS server decreases with an increasing number of subscribers. We observe a received throughput of about 16000 msgs/s for one subscriber and of about 130 msgs/s for 320 subscribers.

Unlike in Figure 3(a) or Figure 3(b), the received message rate decreases significantly with an increasing number of subscribers m. This can be explained as follows. No filters are applied and all messages are delivered to all subscribers such that each message is replicated m times. We call this a replication grade $r = m$. This requires

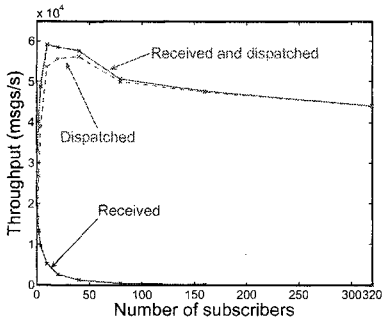


Fig. 4. Impact of the number of subscribers on the message throughput (20 publishers).

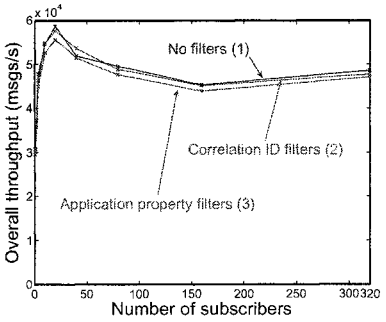


Fig. 5. Impact of filter activation and the number of subscribers on the message throughput.

more CPU cycles for dispatching messages and increases the overall processing time of a single received message. As a consequence the rate of received messages at the server decreases. Thus, the replication grade has to be considered when performance measures from different experiments are compared.

4.2 Impact of Filter Activation

We evaluate the impact of filter activation on the message throughput. We perform 3 different experiment series where all publishers send messages with an application property or correlation ID value set to #0. We install 20 publisher threads on a single publisher machine and a varying number of m subscriber threads on one subscriber machine. We use the following filter configurations which lead to a message replication grade of $r = m$.

(1) No filters are installed.
(2) A filter for #0 is installed by each subscriber as correlation ID.
(3) A filter for #0 is installed by each subscriber as application property.

the number of subscribers on the message

Figure 5 illustrates the overall message throughput for the above described experiments. The overall message throughput differs only slightly for the three different experiments. For other server types, like Bea WebLogic [18] or FioranoMQ [5], filter activation results in a decreased overall throughput compared to experiment (1). A comparison of the results for experiment (2) and (3) shows that correlation ID filters lead to a slightly larger throughput than application property filters for the ActiveMQ server. In the following experiments we focus only on application property filters because they are more flexible.

From Figure 5 we can also conclude, that an increasing number of equal filters has almost no impact on the overall throughput for more than 300 installed filters. Since the overall throughput remains almost constant at a value of 50000 msgs/s.

4.3 Impact of Complex OR-Filters

A single client may be interested in messages with different correlation IDs or application property values. There are two different options to get these messages. The client sets up subscribers

(1) with a simple filter for each desired message type.
(2) with a single but complex OR-filter searching for all desired message types.

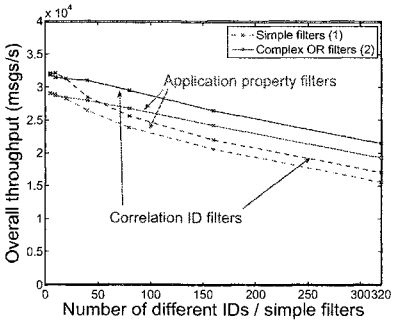


Fig. 6. Impact of simple filters and complex OR-filters on the message throughput for a replication grade of $r=1$.

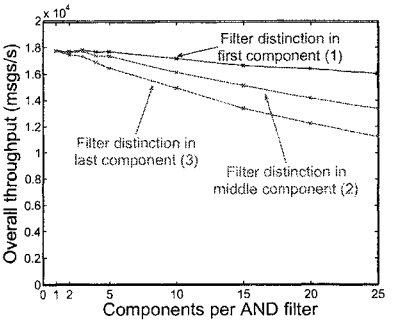


Fig. 7. Impact of an early non-match decision for AND-filters on the message throughput depending on the filter complexity for a replication grade of $r=1$.

These two options are alternative filter configurations for the same application scenario. We assess the JMS server performance for both options by two different experiments, which have both a message replication grade of $r=1$ if the publishers send IDs from #1 to #n in a round robin fashion.

(1) To assess simple filters, we set up for each different ID exactly one subscriber with a filter for that ID.
(2) To assess complex filters, we set up 5 different subscribers numbered from 0 to 4. Subscriber j searches for the IDs $\#(j \cdot \lfloor \frac{n}{5} \rfloor + i)$ with $i \in [1; \frac{n}{5}]$ using an OR-filter.

We use in this experiment one publisher machine with 20 publisher threads and one subscriber machine with a varying number of subscribers for the simple filters approach or 5 subscribers for the complex or filters experiment, respectively. We also repeat the experiment for correlation ID and application property filters.

Figure 6 shows the message throughput depending on on the number of different IDs n in the complex filter. The throughput for the complex OR filters and the simple filters is for both different filter types in the same order of magnitude. The throughput for the simple filter experiment (1) is mostly lower than the throughput for the complex OR filters experiment (2). Also from this experiment we can conclude, that throughput performance of the application property filters and the correlation ID filters only slightly differ.

4.4 Impact of Complex AND-Filters

In the application header section of a message, multiple properties, e.g. $P_1, \ldots, P_k$, can be defined. Complex AND-filters may be used to search for specific message types. In the following, we assess the JMS server throughput for complex AND-filters. They are only applicable for application property filtering. We use one machine with 20 publisher threads and one machine with $m=10$ subscriber threads that are numbered by $j \in [1; m]$. We design three experiment series with a different potential for optimization of filter matching. The subscribers set up the following complex AND-filters of different length n:

(1) for subscriber j:$P_1 = \#j, P_2 = \#0, \ldots, P_n = \#0$

(2) for subscriber j if n is odd:
$P_1 = \#0, \ldots, P_{\frac{n+1}{2}-1} = \#0, P_{\frac{n+1}{2}} = \#j, P_{\frac{n+1}{2}+1} = \#0, \ldots, P_n = \#0$
for subscriber j if n is even
and if $j \leq \frac{n}{2}$: $P_1 = \#0, \ldots, P_{\frac{n}{2}-1} = \#0, P_{\frac{n}{2}} = \#j, P_{\frac{n}{2}+1} = \#0, \ldots, P_n = \#0$
and if $j > \frac{n}{2}$: $P_1 = \#0, \ldots, P_{\frac{n}{2}} = \#0, P_{\frac{n}{2}+1} = \#j, P_{\frac{n}{2}+2} = \#0, \ldots, P_n = \#0$
(3) for subscriber j: $P_1 = \#0, P_2 = \#0, \ldots, P_n = \#j$

The corresponding messages are sent by the publishers in a round robin fashion to achieve a replication grade of $r = 1$. Then, the filters can reject non-matching messages by looking at the first component in experiment (1), at the first half of the components in experiment (2), and at all n components in experiment (3). This experiment is designed such that both the replication grade and the number of subscribers is constant and that only the filter complexity n varies. To avoid any impact of different message sizes in this experiment series, we define $k = 25$ properties in all messages to get the same message length.

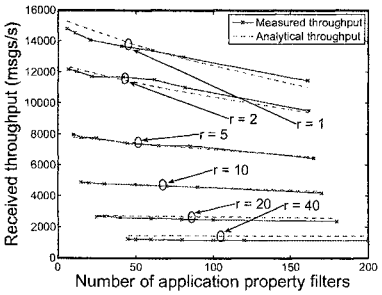
Figure 7 shows the message throughput depending on the filter complexity n. In all scenarios, the filter complexity reduces the server capacity. Experiment (1) yields the largest message throughput, followed by experiment (2) and (3). The evaluation of a complex filter is obviously stopped as soon as a mismatch of one of the components occurs and the evaluation is performed from left to right of its components, as required by the JMS API [19]. precedence level. Parentheses can be used to change this order. This early reject decision of the filters reduces the processing time of a message and increases thereby the server capacity. As a consequence, practitioners should care for the order of individual components within AND-filters: components with the least match probability should be checked first.

4.5 Joint Impact of the Number of Filters and the Replication Grade

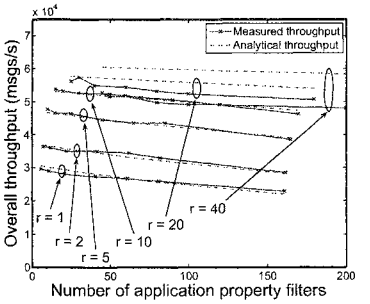
We study the impact of the replication grade r, the number of filters n_{fltr} on the message throughput of the JMS server.

Experimental Analysis The publishers send only messages with ID #0 as a property in the application property part. To achieve a replication grade of r, we set up r different subscribers with a filter for ID #0. Furthermore, we install m_{subs}^{add} additional subscribers, each installing a filter for ID #1. So we have a total of $m_{subs}^{add} + r$ subscribers installed and the number of filters in the system is $n_{fltr} = m_{subs}^{add} + r$. We use the following values for our experiments $r \in \{1, 2, 5, 10, 20, 40\}$, $m_{subs}^{add} \in \{5, 10, 20, 40, 60, 80, 160\}$, and conduct them with 20 publisher threads on one publisher machine and with a variable number of $r + m_{subs}^{add}$ subscribers equally distributed over two subscriber machines.

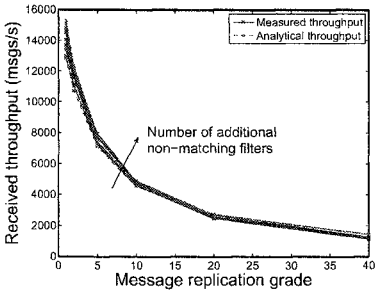
The solid lines in Figure 8(a) and Figure 8(b) show, that the measured received and overall throughput slightly decreases for an increasing number of installed filters n_{fltr} for the above described experiments. The message throughput is also clearly influenced by the message replication grade r. Therefore, we provide in Figure 8(c) and Figure 8(d) an alternative presentation of the same data with the replication grade on the x-axis and separate curves for the number of additional non-matching filters. Figure 8(c) and Figure 8(d) show that the received throughput clearly decreases and the overall throughput clearly increases with an increasing replication grade. Comparing Figure 8(c) and Figure 8(d) with Figure 8(a) and Figure 8(b) leads to the conclusion that the impact of the



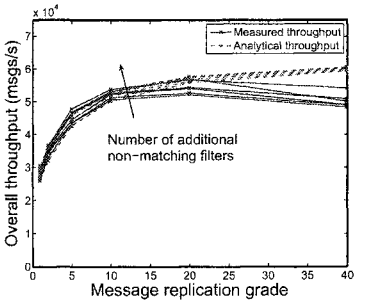
(a) Impact of the number of filters on the received throughput



(b) Impact of the number of filters on the overall throughput



(c) Impact of the replication grade on the received throughput



(d) Impact of the replication grade on the overall throughput

Fig. 8. Impact of the number of application property filters and the replication grade on message throughput of the ActiveMQ server

message replication grade on the message throughput is larger than the impact of the number of installed filters. The throughput is the same regardless if we use equal or different filters that do not match. Which is unlike with the SunMQ [20] which leads to a different analytical model. We conducted the same experiment for correlation ID filters and obtained very similar results.

A Simple Model for the Message Processing Time We assume that the processing time of the JMS server for a message consists of three components. For each received message, there is

- a fixed time overhead t_{rcv} which is almost independent of the number of installed filters.
- a fixed time overhead $n_{fltr} \cdot t_{fltr}^{all}$ caused by the JMS server due to the number of all installed filters. This value depends on the application scenario.
- a variable time overhead $r \cdot t_{tx}$ depending on the message replication grade r. It takes into account the time the server takes to forward r copies of the message.

This leads to the following message processing time B:

$$B = t_{rcv} + n_{fltr} \cdot t_{fltr} + r \cdot t_{tx}. \tag{1}$$

From previous study it is known, that this model holds also for other JMS server implementations, e.g. the FioranoMQ [21] and the Bea WebLogic [18]. Within time B,

one message is received and r messages are sent on average. Therefore, the received and overall throughput are given by $\frac{1}{B}$ and $\frac{r+1}{B}$, respectively. The values for t_{rcv}, t_{fltr}, and t_{tx} can be derived from the measured received message throughput in our experiments by least square approximation.

Validation of the Model by Measurement Data The curves for replication grade $r=20$ and $r=40$ do not follow the trend of the curves for replication grades $r\in\{1,2,5,10\}$. Therefore, we respect only the experiments with replication grades $r\in\{1,2,5,10\}$ in the least squares approximation and obtain the model parameters $t_{rcv}=4.9\cdot10^{-5}$s, $t_{fltr}=1.6\cdot10^{-7}$s, and $t_{tx}=1.5\cdot10^{-5}$s. We use these parameters to calculate the analytical throughput which is illustrated in Figures 8(a)–8(d) by dashed lines. For small replication grades $r=\{1,...,10\}$ the analytical throughput is in good accordance with the measured throughput. We also measured the performance of the correlation ID filters. But we measured only slightly different message throughput values and we can therefore omit a rather complex model as for example used for the SunMQ [20] server. As mentioned above, the capacity curves for message replication grades $r=20$ and $r=40$ in Figure 8(b) are lower than expected from an intuitive extrapolation of the other curves. The reason for this observation might be a maximum internal transmission capacity of the server such that the server CPU is no longer the limiting criterion. In previous studies with other server types we have not encountered such a phenomenon since the overhead of these servers for message filtering was significantly larger than the one for ActiveMQ. As a consequence, the transmission capacity of these servers was sufficient even for a large message replication grade of $r=40$. However, we expect to observe similar saturation effects for these servers, too, if we further increase the message replication grade in this experiment series.

5 Conclusion

In this work, we first gave a short introduction into JMS and reviewed related work. We presented the testbed and explained our measurement methodology before we conducted the experiments. Then we have investigated the maximum message throughput of the Java Messaging System (JMS) server "ActiveMQ" under various conditions.

We studied the server capacity depending on the number of publishers and subscribers and found out that the maximum server performance can be achieved only if sufficiently many publishers and subscribers participate in the system. Correlation ID filters lead to a slightly larger throughput than application property filters and complex OR-filters are more efficient than equivalent single filters. As ActiveMQ obviously supports an early reject for non-matching AND-filters, practitioners should carefully choose the order of the filter components. We showed that the use of filters has only a small impact on the server capacity. This makes the ActiveMQ server very attractive compared to other JMS server solutions [5], [18] if filtering is heavily used. In contrast, the message replication grade has a significant impact on both the received and overall throughput. We finally developed an analytical model to describe the capacity of the ActiveMQ and provided appropriate model parameters based on our measurements. This model is useful to predict the server capacity in practical application scenarios.

Currently, we investigate the capacity of server clusters, which are intended to increase the overall message throughput of the system. Furthermore, we investigate different options to reduce the overhead for filter processing times.

References

1. Sun Microsystems, Inc.: Java Message Service API Rev. 1.1. (2002) http://java.sun.com/products/jms/.
2. Apache: ActiveMQ, Reference Documentation. (2006) http://www.activemq.org.
3. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The Many Faces of Publish/Subscribe. In: ACM Computing Surveys. (2003)
4. Krissoft Solutions: JMS Performance Comparison. Technical report (2004) http://www.fiorano.com/comp-analysis/jms_perf_comp.htm.
5. Fiorano Software, Inc.: FioranoMQ^{TM}: Meeting the Needs of Technology and Business. (2004) http://www.fiorano.com/whitepapers/whitepapers_fmq.pdf.
6. Sonic Software, Inc.: Enterprise-Grade Messaging. (2004) http://www.sonicsoftware.com/products/docs/sonicmq.pdf.
7. Tibco Software, Inc.: TIBCO Enterprise Message Service. (2004) http://www.tibco.com.
8. IBM Corporation: IBM WebSphere MQ 6.0. (2005) http://www-306.ibm.com/software/integration/wmq/v60/.
9. Crimson Consulting Group: High-Performance JMS Messaging. Technical report (2003) http://www.sun.com/software/products/message_queue/wp_JMSperformance.pdf.
10. Sun Microsystems, Inc.: Sun ONE Message Queue, Reference Documentation. (2006) http://developers.sun.com/prodtech/msgqueue/.
11. Carzaniga, A., Wolf, A.L.: A Benchmark Suite for Distributed Publish/Subscribe Systems. Technical report, Software Engineering Research Laboratory, Department of Computer Science, University of Colorado, Boulder, Colorado (2002)
12. Wolf, T.: Benchmark für EJB-Transaction und Message-Services. Master's thesis, Universität Oldenburg (2002)
13. Baldoni, R., Contenti, M., Piergiovanni, S.T., Virgillito, A.: Modelling Publish/Subscribe Communication Systems: Towards a Formal Approach. In: 8th International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2003). (2003) 304–311
14. Baldoni, R., Beraldi, R., Piergiovanni, S.T., Virgillito, A.: On the modelling of publish/subscribe communication systems. Concurrency - Practice and Experience **17** (2005) 1471–1495
15. Mühl, G., Fiege, L., Buchmann, A.: Filter Similarities in Content-Based Publish/Subscribe Systems. Conference on Architecture of Computing Systems (ARCS) (2002)
16. Apache Incubator: ActiveMQ, JMeter Performance Test Tool. (2006) http://www.activemq.org/jmeter-performance-tests.html.
17. Sun Microsystems, Inc.: JRE 1.5.0. (2006) http://java.sun.com/.
18. Bea Systems: Bea WebLogic Server 9.0. (2006) http://dev2dev.bea.com.
19. Sun Microsystems, Inc.: Java Message Service Specification, Version 1.1. (2002) http://java.sun.com/products/jms/docs.html.
20. Henjes, R., Menth, M., Zepfel, C.: Throughput Performance of Java Messaging Services Using Sun Java System Message Queue. In: High Performance Computing & Simulation Conference (HPC&S), Bonn, Germany (2006)
21. Henjes, R., Menth, M., Gehrsitz, S.: Throughput Performance of Java Messaging Services Using FioranoMQ. In: 13thGI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), Erlangen, Germany (2006)

Teil III

Sensornetze und Mobile Ad-Hoc-Netze

Titan: A Tiny Task Network for Dynamically Reconfigurable Heterogeneous Sensor Networks

Clemens Lombriser, Daniel Roggen, Mathias Stäger, Gerhard Tröster

Wearable Computing Lab, ETH Zurich
CH-8092 Zurich, Switzerland
{lombriser,roggen,staeger,troester}@ife.ee.ethz.ch

Abstract. Context recognition, such as gesture or activity recognition, is a key mechanism that enables ubiquitous computing systems to proactively support users. It becomes challenging in unconstrained environments such as those encountered in daily living, where it has to deal with heterogeneous networks, changing sensor availability, communication capabilities, and available processing power.
This paper describes Titan, a new framework that is specifically designed to perform context recognition in such dynamic sensor networks. Context recognition algorithms are represented by interconnected data processing tasks forming a task network. Titan adapts to different context recognition algorithms by dynamically reconfiguring individual sensor nodes to update the network wide algorithm execution.
We demonstrate the applicability of Titan for activity recognition on Tmote Sky sensor nodes and show that Titan is able to perform processing of sensor data sampled at 100 Hz and can reconfigure a sensor node in less than 1ms. This results in a better tradeoff between computational speed and dynamic reconfiguration time.

1 Introduction

Ubiquitous computing systems aim to integrate seamlessly into the environment and to interact with users in unobtrusive ways. As a consequence, the human-computer interface needs to become more intelligent and intuitive [1,2]. One way to achieve this is to make the computing system aware of the context of the user. By knowing what the user is doing, the computing system can proactively support him.

Activities of a person may be recognized using small, wirelessly interconnected sensor nodes worn on the person's body, like motion sensors such as accelerometers integrated in garments or sensors placed in the environment, such as microphones. The sensor nodes must integrate seamlessly into the wearer's clothes. Therefore they need to be of small dimensions and can only offer very limited computational power and memory. Yet activity recognition on such resource-limited devices is still possible using appropriate algorithms [3], which work on an intelligent choice of sensor signal features correlated to the activities to be recognized [4,5].

Context recognition becomes challenging in real-world, unconstrained environments, such as those which are likely to be encountered in daily living situations. In such environments, assumptions on the number and type of sensors available on the body and in the environment are difficult to make. Sensors in the environment may be only available for a short time, e.g. as the user walks by, and context recognition methods must be able to quickly incorporate such temporarily available information. Context recognition gets even more complex when hardware, communication, or power failures are considered.

This paper describes for a system the first time specifically designed to support the implementation and execution of context recognition algorithms in such dynamically changing and heterogeneous environments. We refer to this system as *Titan* – a Tiny Task Network. Titan represents data processing by a data flow from sensors to recognition result. The data is processed by *tasks*, which implement elementary computations like classifiers or filters. The tasks and their data flow interconnections define a task network, which runs on the sensor network as a whole. The tasks are mapped onto the single sensor nodes according to the sensors and the processing resources they provide.

Titan dynamically reprograms the sensor network to exchange context recognition algorithms, handle defective nodes, variations in available processing power, or broken communication links. It has been designed to run on sensor nodes with limited resources such as those encountered in wireless sensor networks.

We have implemented and tested Titan on Tmote Sky [6] sensor nodes and evaluated the performance of the task network execution. The comparison to existing approaches shows that Titan offers a better tradeoff in computational speed vs. dynamic reconfiguration time.

This paper is organized as follows: Section 2 reviews existing approaches for data processing systems in wireless sensor networks. Section 3 describes Titan. It is then evaluated and discussed against a selection of other existing systems in Section 4. In Section 5 and 6 we highlight our future work and eventually conclude this paper.

2 Related Work

An analysis of context recognition methods based on body-worn and environmental sensors was carried out in [7] and has led to the development of the *Context Recognition Network* [8]. It has been designed to run on Linux and to compute context information on a central powerful computer that collects sensor data from the environment. However, this system does not tackle the issue of context recognition in heterogeneous and dynamically organized sensor networks, where the computation is distributed over the whole network.

An approach to dynamic reconfiguration of data processing networks on sensor networks is DFuse [9]. It contains a data processing layer that can fuse data while moving through the network. To optimize the efficiency, the data processing tasks can be moved from one node to another. However, DFuse is targeted at devices with higher processing capabilities than sensor nodes provide.

The *Abstract Task Graph (ATaG)* [10] with its DART runtime system [11] allows to execute task graphs in a distributed manner. The task graph is compiled during runtime and adapted to the configuration of the network. Similar to DFuse, DART imposes too high requirements on the underlying operating system, such that it cannot be run on sensor nodes we want to use.

Dynamic reconfigurability was investigated by providing dynamic loading of code updates in Deluge [12], TinyCubus [13], SOS [14], or [15]. Dynamic code updates however rely on homogeneous platforms (i.e. the same hardware and OS), which is unlikely to be the case in large scale unconstrained networks such as the ones we consider here. In addition, dynamic code loading is time consuming and requires the node to stop operating while the code is uploaded. This may lead to data loss during reconfiguration.

A platform independent approach is to use a virtual machine like Maté [16]. Applications running in Maté use instructions that are interpreted by virtual processors programmed onto network nodes. The performance penalty of the interpretation of the instructions can be alleviated by adding application-specific instructions to the virtual machine [17]. These instructions implement functionality that is often used by the application and execute more efficiently. In contrast to Maté, Titan uses processing tasks as basic building blocks, which include more functionality and thus execute more efficiently.

3 Titan

3.1 Overview

The goal of Titan is to provide a mechanism to dynamically configure a data processing task network on a wireless sensor node network. As the physical location of the sensors on the sensor nodes is important for their contribution to the context recognition application, the individual sensor nodes must be individually reprogrammable.

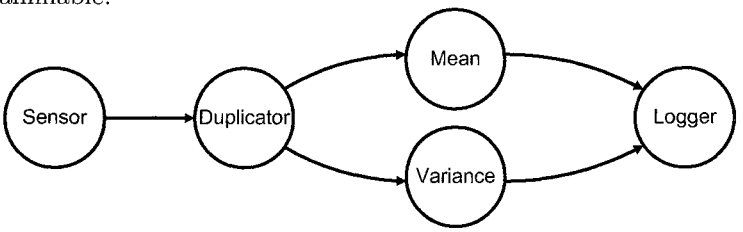


Fig. 1. This task network describes a simple application which reads sensor data, computes two features and stores the results in the Flash memory

Titan provides a set of *tasks*, of which each implements some signal processing function. *Connections* transport the data from one task to another. Together, the tasks and connections form a *task network*, which describes the application to be run on the wireless sensor network. Figure 1 shows an example of such a task network that reads data from a sensor, computes the mean and variance, and stores both values for logging purposes.

Programming data processing in this abstraction is likely to be more intuitive and less error prone than writing sequential code. The inner working of every processing task have to be thoroughly checked only once. This can be done in an isolated way and reduces the complexity of debugging.

Tasks have a set of input ports, from which they read data, process it, and deliver it to a set of output ports. Connections deliver data from a task output port to a task input port and store the data as *packets* in FIFO queues.

The application is issued by a *master node*. It analyzes the task network and splits it into *task subnetworks*, which are to be executed on the individual nodes. The connections between tasks on different nodes is maintained by sending the packets in messages via a wireless link protocol.

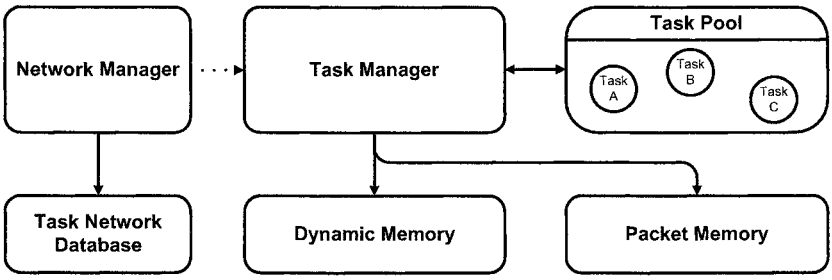


Fig. 2. Main modules of the Titan architecture. The arrows indicate in which direction functions can be called

Titan is built on top of the TinyOS operating system [18], which has been designed to be used on very resource limited sensor nodes. The Titan architecture is shown in Figure 2 and includes the following main modules.

- The **Network Manager** organizes the network to execute the task networks. It connects to the **Task Network Database** where the task network descriptions are located and decides which sensor nodes will execute which part of the task network. These modules are only available on sensor nodes with enough resources to perform the network management.
- The **Task Manager** is the system allowing to reconfigure a sensor node. It maintains a task pool with all tasks available on the sensor node and instantiates the tasks according to the Network Manager's requests. The Task Manager is responsible for reorganizing the task subnetwork executed on the local sensor node during a reconfiguration.
- A **Dynamic Memory** module allows tasks to be instantiated multiple times, and reduces static memory requirements of the implementation. The tasks can allocate memory in this space for their individual state information. This module is needed as TinyOS does not have an own dynamic memory management.
- The **Packet Memory** module stores the packets used by the tasks to communicate with each other. The packets are organized in FIFO queues, from which tasks can allocate packets before sending them. This data space is shared among the tasks.

3.2 Tasks

Titan defines a global set of tasks. Each task implements a data processing algorithm or an interface to the hardware available on the sensor node. Every node in the sensor network implements a subset of all tasks, depending on its processing capabilities. The tasks go through the following phases when they are used:

1. **Configuration** – At this point, the Task Manager instantiates a task. To each task it passes configuration data, which adapts the task to application needs. Configuration data may include sampling frequency, the window size to use, or similar. The task can allocate dynamic memory to store state information.
2. **Runtime** – Every time a task receives a packet, it is allowed to execute to process the data. Titan provides the task with the state information it has set up during the configuration time. Tasks are executed in the sequence they receive a packet, and each task runs to completion before the next task can start.
3. **Shutdown** – This phase is executed when the task subnetwork is terminated on the node. All tasks have to free the resources they have reserved.

3.3 Connections

Packets exchanged between the tasks carry a timestamp and information of the data length and type they contain. Tasks reading the packets can decide on what to do with different data types. If unknown data types are received, they may issue an error to the Task Manager, which may forward it to the Network Manager to take appropriate actions.

To send a packet from one sensor node to another, Titan provides a *communication* task, which can be instantiated on both network nodes to transmit packets over a wireless link protocol as shown in Figure 3. During configuration time the communication task is told which one of its input ports is connected to which output port of the receiving task on the other node. The two communication tasks handle communication details, such as routing or reliable transmission of the packet data in the wireless sensor network. The communication task is automatically instantiated by the Network Manager to distribute a task network over multiple sensor nodes.

The recommended maximum size of a packet for Titan is 24 bytes, as it can easily be fitted with 5 bytes header into a TinyOS active message. The active message is used to transmit data over wireless links and offers 29 bytes of payload.

3.4 Dynamic Allocation of the Task Network

A programmer designs his application by selecting tasks from the task list Titan provides and interconnecting them to form a task network. Task parameters as

well as location constraints can also be defined. The description of the resulting task network is then loaded into the *Task Network Database* in the network.

When the execution of a specific task network is requested, the Network Manager first inspects the capabilities of the sensor nodes in the environment by broadcasting a *service discovery* message containing a list of tasks to be found. Every node within a certain hop-count responds with the matching tasks it has in its Task Pool, and adds status information about itself. Using this information, Network Manager decides whether the task network can be executed, and which node runs which tasks. This is currently done by first placing tasks with location constraints on the first node found with matching requirements. In a second step, the remaining tasks are greedily added to the nodes. If the node capacities are reached, new nodes are added to the processing network until all tasks have been placed.

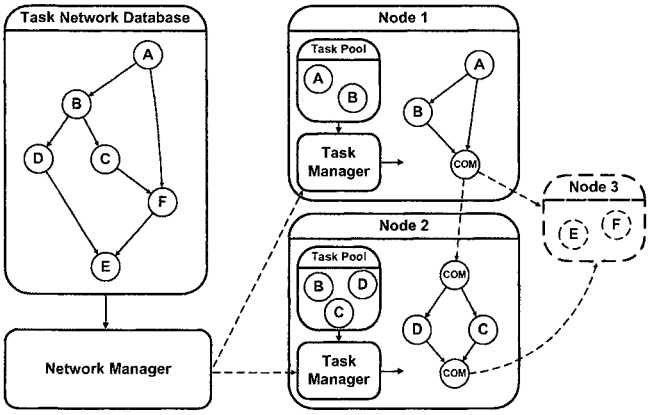


Fig. 3. Allocation of a task network: parts of the task network are configured onto each participating node, depending on their sensors or computational capabilities. Interconnections across sensor nodes are realized over special communication tasks

When data needs to be exchanged across nodes, communication tasks (see Section 3.3) are automatically inserted. The resulting task subnetworks containing the tasks to be executed on every sensor node are then send to each participating node's Task Manager, which takes care of the local instantiation as shown in Figure 3. After the configuration has been issued, the Network Manager keeps polling the Task Managers about their state and changes the network configuration if needed. On node failures, the Network Manager recomputes a working configuration and updates the subnetworks on individual sensor nodes where changes need to be made, resulting in a dynamic reorganization of the network as a whole.

3.5 Synchronization

When sensors are sampled at two sensor nodes and their data is delivered to a third node for processing, the data streams may not be synchronized due to dif-

fering processing and communication delays in the data path. As a consequence, a single event measured at the two nodes can be mistaken for two.

If the two sensor nodes are synchronized by a timing synchronization protocol, a timestamp can be added to the data packet when it is measured. The data streams can then be synchronized by matching incoming packets with corresponding timestamps. Timing protocols have been implemented on TinyOS with an accuracy of a few 10 μs [19,20].

If the two sensor nodes are not synchronized, the sensor data can be examined as in [8]. The idea is to wait until an event occurs that all sensors can measure, e.g. a jump for accelerometers on the body. Subsequent packets reference their timestamp to the last occurrence of the event. This functionality is provided in the *Synchronizer* task.

4 Evaluation

We have implemented and tested Titan on Tmote Sky motes [6]. The memory required by the implementation is listed in Table 1, showing the memory footprints of a plain TinyOS implementation, the virtual machine Maté, the code distribution framework Deluge, and Titan. These numbers change when compiled for different platforms, but give an indication of the size of the Titan implementation.

The space reserved for dynamic and packet memory RAM can be tailored to the needs of the application and the resources on the node. The task number and type in the Task Pool on the node determines the amount of ROM memory requirement, and can be adapted to the platform as well. The memory footprint shown in Table 1 includes all Titan tasks as listed in Table 3.

Platform	ROM	RAM
TinyOS[1]	16520	541
TinyOS with Deluge	26896	1089
Maté	37004	3146
Titan	35024	1422
dynamic memory		+ 4096
packet memory		+ 1440

Table 1. Memory footprints (bytes). The Tmote Sky module provides 48k ROM and 10k RAM.

Interface function	Cycles	Time (μs)
pasteContext	85	16
getContext	145	28
allocPacket	370	70
sendPacket	290	55
hasPacket	25	5
getNextPacket	425	81
Transfer 1 packet (avg)	1026	195

Table 2. Cycle count of the most important Titan interface functions

Table 2 lists the time needed for the most important functions Titan offers to the tasks. All times have been measured by toggling a pin of the microcontroller on the Tmote Sky. The average packet transfer is measured from the point where the sending task calls the sending function to the time where the receiving task has retrieved the packet and is ready to process its contents. This time is roughly

[1] As distributed with Tmote Sky modules, and instantiating the Main, TimerC, GenericComm, LedsC, and ADCC components

$200\mu s$. For the recognition of movements, acceleration data is usually sampled at less than 100 Hz [4], which leaves the tasks enough time for processing.

Table 3 shows the currently available tasks for Titan. The delays given in the table indicate how long each task needs to process a data packet of 24 bytes, which is the recommended Titan packet size as mentioned in Section 3.3.

Name	Descriptions	Delay	RAM	ROM
Duplicator	Copies a packet to multiple output ports	$192\mu s$	–	250
FBandEnergy	Computes the energy in a frequency band from FFT data	$200\mu s$	12	410
FFT	Computes a 32 bit real-valued FFT over a data window of $n = 2^k$ samples (delay for 128 16 bit samples)	$186ms$	$16 + 4n + n_s$	4714
Led	Displays incoming data on the mote LED array	$36\mu s$	–	260
Mean	Computes the mean value over a sliding window of size n	$318\mu s$	$12+n_s$	494
Merge	Merges multiple packets into one	$328\mu s$	12	454
MinMax	Looks for the maximum and minimum in a window of size n	$193\mu s$	8	484
ExpAvg	Computes an exponential moving average over input data	$222\mu s$	8	416
Synchronizer	Synchronizes data by dropping packets until an user defined event occurs	$220\mu s$	10	476
Threshhold	Quantizes the data by applying a user-defined number n of thresholds	$95\mu s$	4+2n	424
TransDetect	Detects value changes in the input signal, and issues a packet with the new value	$201\mu s$	2	474
Variance	Computes the variance over a sliding window of size n	$1510\mu s$	$16+n_s$	720
Zero crossings	Counts the number of zero crossings in the data stream	$176\mu s$	8	370

Table 3. Titan task set and delay T_i on a Tmote Sky. RAM indicates the number of dynamic memory bytes allocated, ROM the bytes of code memory used. The delay has been computed for a packet of 22 bytes data. n_s gives memory bytes needed to store n in the data type used, e.g. for 16 bit values $n_s = 2n$

Most task delays are in the range of a few hundred microseconds, which shows that a task network has enough time to execute when using a sampling frequency of 100 Hz. Even an FFT can be performed over 128 samples in 180ms, leaving 86% of the sample time for processing. Whether a whole task network can process the sampled data in real-time needs further analysis. If the sensor data is sampled with a frequency of f_{ADC} and the recorded samples are issued in packets of N_{ADC} samples, the time left for processing of the local task network is:

$$T_{free}(N_{ADC}, f_{ADC}) = \frac{N_{ADC}}{f_{ADC}} - N_{ADC} \cdot t_{sample} - t_{ADCMsg} \quad (1)$$

Where t_{sample} is the time needed by the sensor to sample one sample, and t_{ADCMsg} the time needed to issue a packet.

The time needed for the processing of the data, i.e. executing the task network is determined by the delays T_i of the allocated tasks with their configuration D_i, and the number N_p of messages exchanged, which needs the time t_p.

$$T_{used}(T) = N_p \cdot t_p + \left(\sum_{\forall i \in T} T_i(D_i) \right) + O(T) \quad (2)$$

The TinyOS scheduler overhead is included by the $O(T)$ function. The execution of the task network is thus feasible if the following inequality holds true:

$$T_{used} < T_{free} \tag{3}$$

In a heterogeneous network, the times needed to execute a task differ from node to node, such that an adaption of the times are needed. This can be done by a simple factor as proposed in [7], where every node indicates a speed grade that is multiplied with the task execution time. A more exact approach would be to store a task execution time table on every node, which the Task Manager uses to determine whether the assigned task network is actually executable.

4.1 Configuration Times

To analyze the time of a reconfiguration, we have configured a node with a task subnetwork containing a counter task that increments every second and sends its data to the LED task, which shows the counter value on the Tmote's LEDs. The task subnetwork description has a size of 19 bytes and fits in a single configuration message. Table 4 shows times needed from the reception of the configuration message to the point where the task subnetwork runs.

Task	Time [μs]
Process configuration message	260
Clearing existing task subnetwork	56
Configuration & Startup	196
Total (with OS overhead)	650

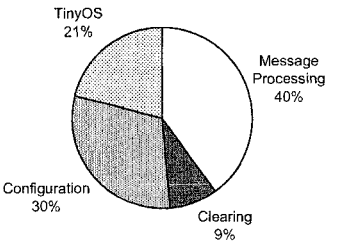


Table 4. Analysis of the reconfiguration process

If a reconfiguration needs multiple configuration messages, Titan stops the current task subnetwork on the reception of the first message. The configuration of the new task subnetwork is then continued every time new configuration packets are received. As soon as the task subnetwork information is complete, Titan starts the execution and notifies the Network Manager of that fact. The continuous processing of the incoming configuration messages reduce the delay after the reception of the last message, as it only includes the configuration and startup time.

4.2 Case Study

To compare the Titan framework to other systems, we have chosen a simple application and have implemented it in three systems: Titan, Maté [16], and Deluge [12]. Maté provides a virtual machine on network motes and thus allows to distribute code in heterogeneous networks, while Deluge allows to reprogram

motes and thus allows to upload platform optimized application-specific code. Thus the comparison involves a general and a specialized solution next to Titan.

The test application continuously samples a sensor at 10 Hz, calculates the maximum, the minimum, and the mean over 10 samples, and sends them to another node. We have measured the number and total size of the configuration messages to be sent, and evaluated how long the processing of the samples takes on the node. The results can be seen in Table 5, respectively in Table 6.

Platform	microseconds
Titan	3.68 ms
Maté	24.0 ms
Deluge	201 μs

Table 5. Processing delay for the case study

Platform	Size [Bytes]	Packets
Titan	71	4
Maté	75	4
Deluge	29588	1345

Table 6. Configuration data size

The numbers show that Titan executes 6.5 times faster than the Maté virtual machine and has a similar configuration size. Deluge on the other hand has an application specific image and is about 18x faster than Titan, but, due to the large number of configuration messages, it needs several seconds for transferring a program image and reboot. This time is not acceptable in the context recognition applications that we envision, where sensors, computational power, and communication channels may change dynamically and in unpredictable ways depending on the user location, motion, social interaction, etc. Deluge does allow to store a certain number of configurations, depending on the node Flash memory, but this allows only a small number of different task sets to execute, while Titan can be reconfigured to a much broader range of applications.

Note that we have chosen a simple application capable of running on Maté. Maté is not able to support sampling rates higher than 10 Hz. It neither can compute a FFT at 10 Hz in realtime, which Titan is able to do. Being able to compute a FFT in realtime is important as many features for activity recognition are gained from the frequency space [3,5].

5 Discussion and Future Work

The work presented here shows that Titan is capable of performing simple context recognition tasks. We are working on extending the Titan task set with classifiers and context recognition algorithms that optimally exploit the sensors on the body and in the environment. These algorithms adapt to the sensors available at the moment and reorganize the computation to changing conditions. This will allow Titan to tackle more complex context recognition tasks.

There are many parameters in the algorithms for the distribution of the task network and the monitoring and failure handling that need to be investigated in more detail. A better approach would use power or transmission cost metrics to arrange tasks on the network. It would also be interesting to enable Titan to dynamically rearrange single tasks during the execution instead of reconfiguring whole nodes if changes to the task network configuration need to be made.

Titan currently only allows to run one configuration at a time. Multiple task networks could be running in parallel and be configured on the nodes separately. This would require the Task Manager to handle accesses to resources that are available only one time on a node and are used with different parameters.

6 Conclusion

We have described the Titan architecture and how it can execute context recognition algorithms in dynamically changing and heterogeneous sensor networks. Titan provides a set of tasks with functionality and interconnects them to form a task network, which describes the data processing algorithm. The task networks are then split into task subnetworks and assigned to the different nodes in a heterogeneous network.

In unconstrained, every-day environments the available sensors, computational power, and communication links may change unpredictably. Titan provides fast reconfiguration and high processing speed, which make it an ideal platform for context recognition in such environments in comparison to virtual machine or dynamic code upload approaches. Titan provides a set of data processing tasks that can be used for various applications and offers easy programming. The task networks are automatically adapted to the sensor node network and are able to compute user context in the network.

The results of the implementation show that a sensor node reconfiguration can be made in less than 1 ms, and sampling rates of 100 Hz can be supported with enough free processing time for recognition algorithms. Thus Titan offers a better tradeoff between processing time and dynamic reconfiguration delay than related approaches.

Acknowledgment: This paper describes work undertaken in the context of the e-SENSE project, 'Capturing Ambient Intelligence for Mobile Communications through Wireless Sensor Networks' (www.ist-e-SENSE.org). e-SENSE is an Integrated Project (IP) supported by the European 6th Framework Programme, contract number: 027227

References

1. Mann, S.: Wearable Computing as Means for Personal Empowerment. In: Proceedings of the 3rd International Conference on wearable Computing (ICWC). (1998) 51–59
2. Starner, T.: The Challenges of Wearable Computing: Part 1 and 2. IEEE Micro **21**(4) (2001) 44–67
3. Stäger, M., Lukowicz, P., Tröster, G.: Implementation and Evaluation of a Low-Power Sound-Based User Activity Recognition System. In: Proceedings of the 8th IEEE International Symposium on Wearable Computers (ISWC). (2004) 138–141
4. Huynh, T., Schiele, B.: Analyzing features for activity recognition. Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies (2005) 159–163

5. Bharatula, N.B., Stäger, M., Lukowicz, P., Tröster, G.: Empirical Study of Design Choices in Multi-Sensor Context Recognition Systems. In: Proceedings of the 2nd International Forum on Applied Wearable Computing (IFAWC). (2005) 79–93
6. Moteiv Corporation: Tmote Sky: http://www.moteiv.com (2005)
7. Anliker, U., Beutel, J., Dyer, M., Enzler, R., Lukowicz, P., Thiele, L.: A Systematic Approach to the Design of Distributed Wearable Systems. IEEE Transactions on Computers **53**(8) (2004)
8. Bannach, D., Kunze, K., Lukowicz, P., Amft, O.: Distributed Modular Toolbox for Multi-modal Context Recognition. In: Proceedings of the 19th International Conference on Architecture of Computing Systems (ARCS). (2006) 99–113
9. Kumar, R., Wolenetz, M., Agarwalla, B., Shin, J., Hutto, P., Paul, A., Ramachandran, U.: DFuse: A Framework for Distributed Data Fusion. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys), New York, NY, USA, ACM Press (2003) 114–125
10. Bakshi, A., Prasanna, V.K.: Programming Paradigms for Networked Sensing: A Distributed Systems' Perspective. In: 7th International Workshop on Distributed Computing (IWDC). (2005)
11. Bakshi, A., Pathak, A., Prasanna, V.K.: System-level Support for Macroprogramming of Networked Sensing Applications. In: Proceedings of the International Conference on Pervasive Systems and Computing (PSC). (2005)
12. Hui, J.W., Culler, D.: The dynamic behavior of a data dissemination protocol for network programming at scale. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, ACM Press (2004) 81–94
13. Marron, P.J., Lachenmann, A., Minder, D., Hahner, J., Sauter, R., Rothermel, K.: TinyCubus: A Flexible and Adaptive Framework Sensor Networks. Proceeedings of the Second European Workshop on Wireless Sensor Networks (2005) 278–289
14. Han, C., Kumar, R., Shea, R., Kohler, E., Srivastava, M.: A Dynamic Operating System for Sensor Nodes. In: 3rd International Conference on Mobile Systems, Applications, and Services. (2005) 163–176
15. Dulman, S., Havinga, P.: Architectures for Wireless Sensor Networks. In: Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). (2005) 31–38
16. Levis, P., Culler, D.: Maté: A Tiny Virtual Machine for Sensor Networks. ACM SIGOPS Operating Systems Review **36**(5) (2002) 85–95
17. Levis, P., Gay, D., Culler, D.: Active Sensor Networks. In: Proceedings of the 2nd USENIX/ACM Symposium on Network Systems Design and Implementation (NSDI). (2005)
18. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for network sensors. In: Architectural Support for Programming Languages and Operating Systems. (2000)
19. Elson, J., Girod, L., Estrin, D.: Fine-grained network time synchronization using reference broadcasts. In: Proceedings of Fifth Symposium on Operating Systems Design and Implementation (OSDI). (2002) 147–163
20. Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-sync Protocol for Sensor Networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems. (2003) 138 – 149

Key Exchange for Service Discovery in Secure Content Addressable Sensor Networks

Hans-Joachim Hof, Ingmar Baumgart, and Martina Zitterbart

Institute of Telematics, Universität Karlsruhe (TH), Germany
{hof,baumgart,zit}@tm.uka.de

Abstract. *Secure Content Addressable Network (SCAN)* is an architecture for service discovery in service centric sensor networks that enables dynamic service composition. This paper proposes two new security mechanisms for SCAN: *Single Path Key Exchange (SPX)* and *Multi Path Key Exchange (MPX)*. Both security mechanisms allow two arbitrary nodes of SCAN to exchange a symmetric key for secure communication. We also propose to use replication service information and majority vote to achieve security.
We evaluated the performance and security of Secure Content Addressable Networks with Single Path Key Exchange, Multi Path Key Exchange and replication using a worst case attack model. It has been found, that in a network with 1000 nodes and 5% malicious nodes the probability of a successful lookup operation is still 80%. The results of the simulation indicate, that the overhead and the security level of SCAN with SPX and MPX scale with an increasing number of nodes. The simulation results also show that SCAN is suitable for networks with 100 to 1000 nodes.

1 Introduction

Sensor networks, as we consider them, are resource constrained with respect to memory and computing power of the sensor nodes. Therefore, public key cryptography is not possible because it involves a lot of computing power. We also expect that there is no infrastructure like a public key infrastructure or some powerful server which is available all the time. We focus on scenarios like assisted living, health care, and home automation. In these scenarios, hundreds or thousands of nodes can be in use and the density of nodes in the network is typically high. New services can be dynamically created during the lifetime of the network. To exploit the full power of dynamic service composition, nodes need a way to find available services and to discover the properties of these services (e.g. the address or the position).

Secure Content Addressable Network (SCAN) [1][2][3] is an architecture for secure service discovery in sensor networks that allows for dynamic service composition. In this paper, we propose two new security mechanisms for SCAN: *Single Path Key Exchange (SPX)* and *Multi Path Key Exchange (MPX)*. The feasibility of both mechanisms for service discovery in sensor networks is evaluated by simulation.

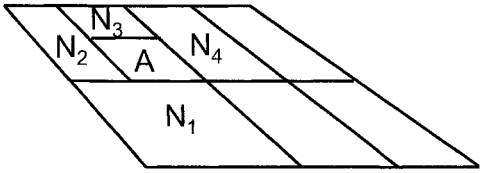


Fig. 1. Example of an unfolded 2-dimensional SCAN space, showing the four neighbor zones (N_1, N_2, N_3, N_4) of a zone A.

2 Secure Content Addressable Network

This section gives an overview of Secure Content Addressable Networks (SCAN). A more detailed description of SCAN can be found in [1], [2], and [3].

Secure Content Addressable Network is based on Content Addressable Network (CAN) [4], which is an overlay network implementing a distributed hash table. SCAN uses a logical virtual d-dimensional coordinate space on a d-torus to store *(service name,service description record)*-pairs. This space is called SCAN space in the rest of this paper. The *service name* is mapped on a coordinate in SCAN space by using a hash function on the service name. The corresponding hash value is interpreted as coordinate in the d-dimensionial coordinate space, e.g. in the case of $d = 2$ the hash value is split into two equal-sized parts x and y and (x, y) is the corresponding coordinate in SCAN space.The *service description record (SDR)* is stored at this coordinate. Service description records hold information about a service, e.g. the network layer address of the service provider. The SCAN space is divided into zones, each owned by one node of SCAN. Hence, if a service description record is stored at a coordinate in SCAN space, the corresponding zone owner stores the service description record. Fig. 1 shows a zone A and its neighbors N_1, N_2, N_3, and N_4 in a 2-dimensional SCAN space.

The secure join operation is used to securely integrate new nodes into a SCAN. During the join operation, the joining node gets assigned a part (zone) of the SCAN space. Each SCAN node maintains a list of network layer addresses of its neighbors in the SCAN space. To avoid that communication between two SCAN neighbors can be attacked, *symmetric keys between SCAN neighbors* are established during the join operation. These symmetric keys can be used to protect the integrity and confidentiality of messages between overlay neighbors hop-by-hop.

SCAN allows for routing in the SCAN space: a SCAN node forwards a message to the SCAN neighbor that is in SCAN space closest to the destination. Nodes that want to retrieve service description records (SDRs) for a specific service compute the hash value of the service name, interpret it as a coordinate in the SCAN space, and send a request message to the calculated coordinates in the SCAN space.

For a d-dimensional SCAN space with N nodes the average routing path length h is:

$$h = \frac{d}{4} N^{\frac{1}{d}} \tag{1}$$

See [4] for details. In section 3 we will show, how the path length h is related to the security of our proposed key exchange protocols.

3 Single Path Key Exchange and Multi Path Key Exchange

When a node stores a (service name, service description record)-pair in SCAN, the message is sent from SCAN node to SCAN node using the routing in SCAN space. Each SCAN node encrypts the service description record (SDR) for its neighbor node. The response is sent in the same way. However, there is no end-to-end encryption between a node A that wants to insert a SDR and a node B that stores that SDR, because A and B have no symmetric key in common. In SCAN, symmetric keys exist only between neighbors, but A and B need not to be neighbors. If A and B have a symmetric key in common (and the network layer address of B is known), it is not necessary to send the service description record using the routing protocol in SCAN space, but it can be sent on network layer. As one hop in the SCAN space can be multiple hops on the network layer, communication on network layer is more efficient than communication in SCAN space. Hence, a key exchange protocol is needed. If a key is exchanged between A and B, updates of service description records are also more efficient. Updates are necessary in SCAN because SCAN requires every service description record to be updated on a regular basis to deal with node failure. If a key exists, the update message can be sent on message layer because integrity and confidentiality of the message can be protected with the key. Otherwise, the update message must be sent in SCAN space.

We propose two key exchange protocols: *Single Path Key Exchange* (SPX) and *Multi Path Key Exchange* (MPX).

3.1 Single Path Key Exchange

If node A uses *Single Path Key Exchange* it creates a symmetric key and sends the key in plain text in the overlay to node B (see Fig. 2). Whenever a node of the overlay forwards a message to one of its neighbors, it uses the corresponding symmetric neighbor key to encrypt the message. This overlay hop-by-hop encryption is possible because symmetric keys between neighbors have been established during the join operation.

For efficiency reasons, the reply of node B is not sent over the overlay but on network layer. Node B sends an encrypted acknowledge. Only after this successful key exchange node A encrypts the SDR and sends it to node B on the

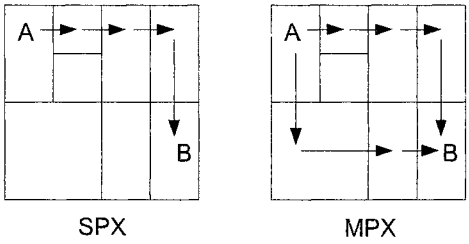


Fig. 2. Single Path Key Exchange (SPX) and Multi Path Key Exchange (MPX)

network layer. Node B uses the exchanged key to decrypt the message and stores the Service Description Record.

Using SPX results in only marginal communication overhead, because only the key is sent over the overlay. The SDR, which is usually larger than a key, is sent directly on the network layer. Hence, this procedure produces less overhead compared to the original SCAN insert operation. Nodes may store the exchanged keys for later updates of the SDRs. As SDRs are stored soft-state, regular updates are necessary. If a key exists, it is no longer necessary to use the overlay for secure communication but the more efficient communication on network layer can be used.

During the key exchange each node on the overlay path between node A and node B can read the symmetric key. Thus these nodes will be able to perform a man-in-the-middle attack on the network layer to tamper with the SDR. This manipulation can not be detected. To accomplish this attack, it is necessary, that the eavesdropper is on the overlay path between node A and node B.

The probability that an overlay path is free of malicous nodes, if the average path length is h and a fraction of m of all nodes in the network are malicious, is $(1-m)^h$.

Thus the probability p of a successful key exchange in a d-dimensional SCAN with N nodes using equation (1) is:

$$p = (1-m)^{\frac{d}{4}N^{\frac{1}{d}}} \tag{2}$$

Single Path Key Exchange can also be used when a node wants to retrieve service descritption records of a service: SPX is executed with the coordinate that is calculated by using a hash function on the service name. The node which stores the service desctiption record then sends back all matching service description records encrypted with the exchanged key on network layer. In this case, SPX is of great use because the list of service description records is much larger than the key, and with SPX, this list need not be transfered in SCAN space but on the more efficient network layer.

3.2 Multi Path Key Exchange

To increase the probability of a successful insert operation in presence of malicious nodes, *Multi Path Key Exchange* uses during the key exchange different paths in the overlay and sends only parts of the key along each overlay path (see also fig. 2). A similar approach has been proposed by [8] for communication on the network layer, but we are, to our best knowledge, the first to apply the idea to a structured overlay network. If node A uses MPX, it creates a symmetric key, splits it into n parts and sends each part in plain text along one of n paths. A very simple approach to split a key into parts is to randomly choose $n-1$ key parts and calculate the last key part by using the XOR-operation ($\oplus$) on the preceding $n-1$ key parts and the key itself:

$$part_n = part_1 \oplus part_2 \oplus ... \oplus part_{n-1} \oplus key$$

We use the fact that in a SCAN with dimension d exist with high probability d nearly optimal distinct paths between two arbitrary nodes because of the structure of the SCAN space. Node B reconstructs the key and sends an encrypted acknowledge message back to node A. In the example above, the key is reconstructed by simply using the XOR-operation on all received keys:

$$key = part_1 \oplus part_2 \oplus ... \oplus part_{n-1} \oplus part_n$$

Node A then encrypts the SDR and sends it to node B on the network layer. To manipulate a SDR without getting noticed, the cooperating attackers must be on each overlay path between node A that uses the insert operation and node B that will store the SDR. More advanced secret sharing schemes [9] can be used so that the key can be reconstructed with only k out of n key parts. These schemes avoid that a single attacker on one overlay path between A and B can hamper the key exchange by inserting fake key parts for a denial-of-service attack. The probability p of a successful key exchange with MPX in a d-dimensional SCAN with N nodes is:

$$p = \sum_{i=k}^{n} \binom{n}{i} \left((1-m)^{\frac{d}{4}N^{\frac{1}{d}}}\right)^{i} \left(1-(1-m)^{\frac{d}{4}N^{\frac{1}{d}}}\right)^{n-i} \tag{3}$$

MPX can also be used to authenticate if a node is at a certain coordinate in the SCAN space: only the node at the destination coordinates of MPX in SCAN space can legitimately receive all the key parts which are sent by MPX. Hence, the knowledge of the exchanged key proves, that the node is really at that coordinate in SCAN space. Thus, if MPX is used for the insert operation, attackers can only claim to be at a fake coordinate in SCAN space if they are on each overlay path and if they cooperate. Hence, a node which uses the insert operation to store a SDR can after MPX be sure with high probability, that it is talking to the node that is expected to store the SDR.

Multi Path Key Exchange can also be used when retrieving service description records. The mechanism is the same as with Single Path Key Exchange (see above).

4 Security by replication of service information

Redundancy can be used to secure the integrity of service description records (SDRs). If a node stores an SDR at different locations of the SCAN space and nodes use multiple of these locations to retrieve service description records, an attacker needs to attack the majority of the SDRs to achieve a high probability of success. The coordinates, at which an SDR is stored, are determined using a hash function. The hash value of the service name is interpreted as a coordinate in SCAN space and the SDR is stored at the node that owns the corresponding zone. One way to store an SDR on multiple nodes is to use multiple hash functions in the computation of the location. If a hash algorithm h is given, several hash functions $(h_1, h_2, h_3, \ldots)$ can be constructed by simply concatenating (|) a number to the string that will be hashed:

$$h_1(value) = h(value|1), h_2(value) = h(value|2), \ldots$$

5 Simulation

To evaluate the feasibility of using an overlay for service discovery in sensor networks and to evaluate the performance of the overlay, we implemented SCAN in the network simulator GloMoSim [10].

5.1 Simulation Settings

The following simulation settings are used for the simulation experiments: the simulation area is 500x500 meters. In this area, 100, 250, and 1000 nodes are placed randomly. The 802.11 MAC layer protocol of GloMoSim is used with a communication range of 158m. These simulation settings are similar to the settings used in other papers. The nodes join the network in constant intervals (120s). Hence, the total simulation time is $number_of_nodes * 120$s.

To simulate the service centric sensor network, each node randomly chooses how many services it will offer (0 to 7 services) and how many services it will use (0 to 10 services) before the node joins the network. A total of 100 different services is present in the network. After joining the network, a node registers all the services that it offers. Later, it searches for the services that it needs. Every 30 minutes of simulation time, a node will re-register its Service Description Records and it will call the lookup operation again for each service it uses. Every second of simulation time, nodes fail with a certain probability. With the same probability, a node searches again for a service that it uses.

The simulation uses static routing with a predefined loss rate. This was the only efficient way to simulate sensor networks with a huge number (> 1000) of nodes in GloMoSim as it turned out that the implementations of routing protocols for GloMoSim do not scale well with the number of nodes. However, this does not affect the conclusion about the simulation results because the underlying routing protocol does not have a high impact on the overlay as long

as the network does not get partitioned. We expect a huge density of sensors, so it is very unlikely that the network gets partitioned. For each combination of parameters several runs with different seeds were done.

5.2 Attack Model

The attack model states "worst case" attackers: it is assumed that all malicious nodes of the network cooperate and that nodes get compromised after being deployed. The probability of a node to get compromised is identical for all nodes of the network. Hence, over time more and more nodes get malicious and the attackers get more powerful. The attackers do not show any suspicious behaviour to their neighbors and they are conforming to the protocol most of the time. So, attackers can not be detected until they start the attack. The simulation marks every message as compromised that passes a malicious overlay node. The number of malicious nodes on the communication paths between two nodes is used to determine success or failure of any operation.

6 Results

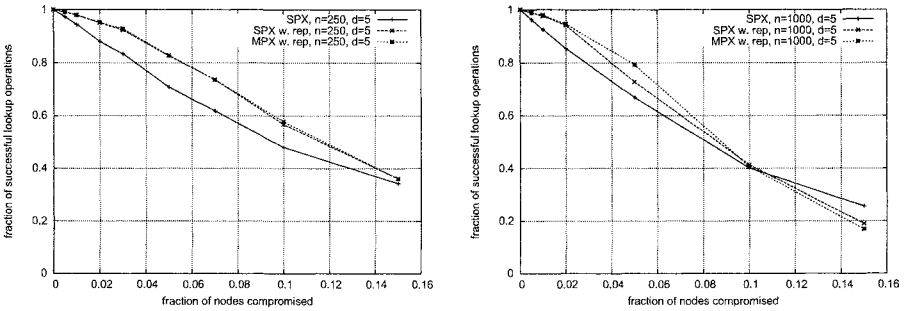
This section presents some simulation results of the GloMoSim implementation of SCAN focusing on security and communication overhead.

6.1 Successful lookup operations

If a single call of the lookup operation returns the Service Description Records of a specific service, the lookup operation is successful. The lookup success is evaluated separately for each node that stores a SDR. If, for example, ten nodes store one Service Description Record each and the lookup operation retrieves only eight of these SDRs, the probability of a successful lookup is 80%. In many scenarios, only one of a number of similar services is really needed, hence the lookup operation would be considered successful by the user, if at least one Service Description Record is retrieved. In such scenarios, the proposed architecture does perform significantly better than presented here. However, we decided to use the more strict definition of lookup success as described above, so the results presented in this paper should be viewed as "worst case" results.

Fig. 3(a) shows the probability of a successful lookup in a network with 250 nodes and a SCAN dimension of 5. For the secure insert operation, we used the methods described in sections 3 and 4: Single Path Key Exchange (SPX), Single Path Key Exchange with replication of SDRs (SPX+Rep), and Multi Path Key Exchange with replication of SDRs (MPX+Rep). For the secure lookup operation, we used SPX without replication of SDRs if no replicates were used by the insert operation; otherwise, we use SPX with replication of SDRs.

SPX with replication of SDRs and MPX with replication of SDRs offer a higher lookup probability than SPX without replication of SDRs. MPX with replication of SDRs has nearly the same probability for a successful lookup than



(a) $n = 250$ nodes (b) $n = 1000$ nodes

Fig. 3. Successful lookup operations in a network with (a) 250 and (b) 1000 nodes, different key exchange methods (SPX, MPX), and with and without replication of Service Description Records.

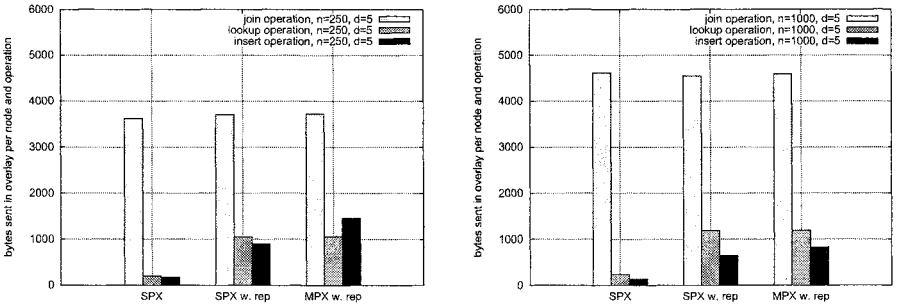
SPX. For example, if 5% of all nodes (=13 nodes) are malicious, we could still retrieve 83% of all Service Description Records. Similar probabilities were found in a network with 1000 nodes: with 5% malicious nodes (=50 nodes) it is still possible to retrieve about 80% of all Service Description Records (see Fig. 3(b)). In the simulation with 1000 nodes, MPX with replication of Service Description Records produces a better possibility of a successful lookup, unless the fraction of malicious nodes is higher than 10%. In this case the majority of involved lookup paths is malicious, thus the majority vote fails. Because MPX without replication performs similarly to SPX with replication, we omit the MPX results in the plots.

Fig. 3 shows that the use of replicates of Service Description Records significantly enhances lookup probability whereas the use of MPX+Rep has an impact only in the simulation with 1000 nodes. The reason for this is the small number distinct paths in small networks.

The simulation results concerning the probability of a successful lookup show that it is possible to retrieve a reasonable number of SDRs even if a moderate fraction of nodes is malicious. In sensor networks, it is often only needed to find only one out of many identical services. Here, SCANs are an ideal solution. The results also show, that SPX with replication of the SDRs should be used by the secure insert operation and the secure lookup operation. MPX should be used only in networks with many nodes. However, MPX offers authentication of a node's coordination in SCAN. SPX does not offer this feature.

6.2 Overhead

In the following, we concentrate on the communication overhead on the overlay layer. Fig. 4 shows this overhead per operation (join, lookup, and insert) and node.



(a) $n = 250$ nodes (b) $n = 1000$ nodes

Fig. 4. Overhead on overlay layer of the join operation, lookup operation, and insert operation with a SCAN dimension of $d = 5$ in a network with (a) 250 and (b) 1000 nodes.

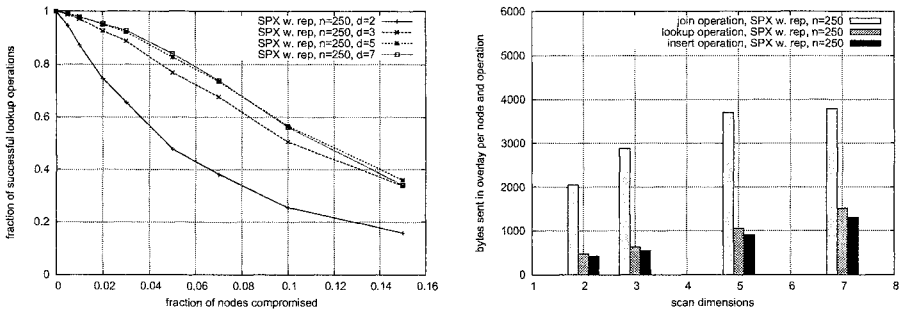
It is clear that the use of replicates multiplies the communication overhead of the lookup operation and of the insert operation because the operations exchange keys for all destinations. MPX with replication produces about a third more traffic than SPX with replication. If we compare Fig. 4(a) and Fig. 4(b) we see that the overall communication overhead only slightly increases with a higher number of nodes.

The simulation results concerning the communication overhead of SCANs show, that the join operation is costly whereas the insert and lookup operation have moderate costs. Although the join operation looks expensive compared to the lookup and insert operation, the join communication costs are incurred only once in the lifetime of a sensor node. In contrast each node performs several hundred lookup and insert operations. Consequently the join operation poses only a small fraction of the total communication overhead.

Because SCAN is an overlay, one hop in SCAN space typically involves multiple hops on the network layer. In our simulation, one overlay hop involved on average 2-3 underlay hops. Thus, to get the total communication overhead on network layer the costs shown in Fig. 4 have to be multiplied by this factor. The resulting costs for insert and lookup operations look very promising for networks with up to 1000 nodes.

6.3 Influence of system parameter

The dimension d of the SCAN space is a parameter of SCAN. It can be chosen freely. However, if we increase d, we also increase the memory usage of every node, because more dimensions result in more neighbors and thus larger neighbor tables must be stored. The neighbor tables store information about the neighbors, e.g. the network layer address and a symmetric key for each neighbor. In SCAN, each node has in average $2d$ neighbors. The advantage of an increased d is, that the average path length between two arbitrary nodes in the overlay



(a) Successful lookup operations. (b) Overhead.

Fig. 5. Influence of SCAN dimension d on successful lookup operations (a) and overhead (b).

decreases, because each node has a higher connection degree (more neighbors). Thus, the probability to run across a malicious node on this path is reduced. As one hop in the overlay usually corresponds to multiple hops on network layer, communication overhead can also be significantly reduced by a higher dimension d. This makes a trade-off between memory and communication overhead possible with SCAN.

We studied how a change of d affects the probability of a successful lookup due to shorter overlay paths and a larger number of overlay neighbors. Fig. 5(a) shows how d affects successful lookup operations in a network with 250 nodes and SPX with replication of Service Description Records used by the insert operation and the lookup operation. We use d replicates of the SDRs. A dimension of $d = 5$ seems to be the ideal choice for the given scenarios (100 to 1000 nodes) as a dimension d of seven does not increase the probability of a successful lookup, because there are not enough nodes in the network to further increase the number of SCAN neighbors. However, this statement may not hold for other numbers of nodes. Fig. 5(b) shows how costly an increase of d is in the same scenario. The increase in replicates causes an increase in communication overhead.

7 Related Work

Several protocols and architectures for service discovery exist. Popular architectures and protocols in infrastructure based networks include e.g. the Service Location Protocol [11] or the Secure Service Provision Protocol [15].

Architectures that use an infrastructure of any kind (e.g., a central server or a public key infrastructure) are not suitable for sensor networks as we see them (see Introduction). The security concepts of service discovery protocols like the Secure Service Provision Protocol, the Secure Service Discovery Protocol [13], or the Secure Service Discovery Protocol [14] are based either on public key

cryptography or preshared secrets (passwords). Both concepts are not suitable for the sensor networks we consider: public key cryptography is at the moment computationally too expensive and preshared secrets are difficult to setup and do not scale well.

There are several methods for key exchange: Diffie-Hellman [16] is computationally too complex for the sensor nodes that we consider. Promising key exchange methods for sensor networks are random-key predistribution protocols, e.g., [17]. Random-key predistribution protocols assign each sensor a random subset of keys out of a very large reservoir of keys. If two nodes want to communicate and they have a key in common, they can use this key. Otherwise, neighbors are used to construct a key. This idea is extended in [8] by using multiple redundant paths to increase the security of the exchanged keys.

8 Conclusion

This paper presented two new security mechanisms for Secure Content Addressable Networks: Single Path Key Exchange (SPX) and Multi Path Key Exchange (MPX). Both mechanisms allow for secure insert and lookup of Service Description Records and both mechanisms allow for more efficient subsequent updates of Service Description Records. MPX also allows for authentication of the node which stores the Service Description Records.

The paper also presented a simulation of Secure Content Addressable Networks with SPX and MPX. The results show that Secure Content Addressable Networks with SPX and MPX provide a reasonable level of security for service centric sensor networks. If, for example, in a network with 1000 nodes 5% of all nodes are malicious, 80% of all lookups are still successful. The results indicate, that Secure Content Addressable Networks with SPX and MPX scale with an increasing number of nodes concerning security level and overhead. The achieved security level can easily be adapted by carefully choosing the dimension d of Secure Content Addressable Networks and the number of replicates at the cost of an increased communication overhead. The simulation results show that Secure Content Addressable Networks are suitable for networks with 100 to 1000 nodes.

References

1. H.-J. Hof, E.-O. Blass, T. Fuhrmann, and M. Zitterbart, "Design of a secure distributed service directory for wireless sensor networks," First European Workshop on Wireless Sensor Networks, Berlin, Germany, Jan. 2004.
2. H.-J. Hof, E.-O. Blass, and M. Zitterbart, "Secure overlay for service centric wireless sensor networks," First European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004), Heidelberg, Germany, Aug. 2004.
3. H.-J. Hof and M. Zitterbart, "SCAN: A secure service directory for service-centric wireless sensor networks," *Computer Communications*, July 2005.
4. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," ACM SIGCOMM 2001, San Diego, California, USA, Aug. 2001.

5. J. R. Douceur, "The sybil attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems.* London, UK: Springer-Verlag, 2002.
6. F. Stajano and R. J. Anderson, "The resurrecting duckling: Security issues for adhoc wireless networks," in *Proceedings of the 7th International Workshop on Security Protocols.* London, UK: Springer-Verlag, 2000, pp. 172–194.
7. D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," Symposium on Network and Distributed Systems Security (NDSS'02), San Diego, California, USA, Feb. 2002.
8. H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," 2003 IEEE Symposium on Security and Privacy, Oakland, California, USA, May 2003.
9. A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, 1979.
10. X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for parallel simulation of large-scale wireless networks," Workshop on Parallel and Distributed Simulation, Banff, Alberta, Canada, 1998.
11. E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2 ," RFC 2608 (Proposed Standard), June 1999, updated by RFC 3224. [Online]. Available: http://www.ietf.org/rfc/rfc2608.txt
12. S. Czerwinski, B. Zhao, T. Hodes, A. D. Joseph, and R. H. Katz, "A secure service discovery service," ACM/IEEE International Conference on Mobile Computing and Networks (Mobicom 1999), Seattle, Washington, USA, Aug. 1999.
13. F. Almenárez and C. Campo, "Spdp: A secure service discovery protocol for adhoc networks," 9th Open European Summer School and IFIP Workshop on Next Generation Networks, Balatonfured, Hungary, Sept. 2003.
14. Y. Yuan and A. William, "A secure service discovery protocol for manet," 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2003), Beijing, China, Sept. 2003.
15. R. Handorean and G.-C. Roman, "Secure service provision in ad hoc networks," First International Conference on Service-Oriented Computing, Trento, Italy, Dec. 2003.
16. W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.
17. L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," Proceedings of the 9th ACM conference on Computer and communications security, Washington, DC, USA, 2002.

NIDES: Ein Verfahren zur Multihop-Distanzschätzung mittels Nachbarschaftsanalyse

Carsten Buschmann, Christian Werner, Horst Hellbrück und Stefan Fischer

Institut für Telematik, Universität zu Lübeck, Ratzeburger Allee 160, 23538 Lübeck

Zusammenfassung. Die Bestimmung räumlicher Entfernungen zwischen einzelnen Knoten ist ein wichtiger Aspekt in drahtlosen Sensornetzwerken. In diesem Beitrag präsentieren wir ein neuartiges Verfahren zur Abstandsschätzung, welches auf dem Vergleich von Nachbarschaftslisten basiert. Es nutzt die Tatsache aus, dass zwei dicht zusammen liegende Knoten mehr gemeinsame Nachbarn haben als solche, die weiter voneinander entfernt liegen. Das Verfahren bietet zwei besondere Vorteile gegenüber anderen Ansätzen. Zum einen kommt es gänzlich ohne zusätzliche Hardware aus und zum anderen basiert es nicht auf der Messung physikalischer Größen, wie etwa der Funksignalstärke, welche durch Umwelteinflüsse oft stark verfälscht werden. Wir erläutern die grundlegende Funktionsweise dieses Schätzverfahrens zwischen direkt benachbarten Knoten und präsentieren anschließend ausführliche Untersuchungen zur Anwendung dieser Technik über mehrere Hops hinweg. Anhand von Simulationsergebnissen demonstrieren wir, dass diese Art der Abstandsschätzung auch im Multihopfall zuverlässige Ergebnisse liefert.

1 Einleitung

Leistungsfähige Techniken zur Selbstlokalisierung sind ein wichtiger Teilaspekt bei der Realisierung selbstorganisierender Systeme [1]: Ein von einem Sensorknoten detektiertes Ereignis ist oft nur zusammen mit der zugehörigen geographischen Position eine "wertvolle" Informationseinheit. Ein Beispiel hierfür ist ein Sensornetzwerk zur Früherkennung von Waldbränden. Detektieren einzelne Knoten einen plötzlichen Temperaturanstieg oder starke Rauchentwicklung, so ist es zwingend erforderlich, diese Ereignisse mit geographischen Positionen zu verknüpfen. Nur so ist es möglich, die Gefahrenzone zu erkennen und dort den Brand zu bekämpfen.

Leider sind GPS-Empfänger zur Positionsbestimmung in Sensornetzwerken unvorteilhaft. Diese Geräte sind teuer und haben eine hohe Leistungsaufnahme; damit widerspricht ihr Einsatz den wesentlichen Grundprinzipien bei der Konstruktion von Sensorknoten: geringer Preis, kleine Bauform und hohe Energieeffizienz. Daher wurden Techniken entwickelt, um die räumliche Position einzelner Knoten auch ohne den durchgängigen Einsatz von GPS-Empfängern bestimmen zu können. Die hier zugrunde liegenden Algorithmen gehen davon aus, dass einige Knoten im Netzwerk ihre geographische Position bereits kennen - dies sind

so genannte Ankerknoten. Alle anderen Knoten versuchen dann ihren Abstand zu mehreren dieser Ankerknoten zu bestimmen und berechnen anschließend mit Hilfe von Multilateration ihre eigene Position. Somit ist die Bestimmung von Abständen zwischen Knoten ein ganz wesentlicher Schritt bei der Positionsbestimmung [11].

Im Rahmen dieses Beitrags präsentieren wir eine erweiterte Form des *Neighborhood Intersection Distance Estimation Scheme (NIDES)*, welches wir in einer ersten Version bereits in einer früheren Arbeit [5] vorgestellt haben. Das besondere bei diesem Schätzverfahren ist, dass es zum einen gänzlich ohne zusätzliche Hardware auskommt und zum anderen sehr robust gegenüber Umwelteinflüssen ist. Bei diesem Schätzverfahren bestimmten zwei Knoten ihren Abstand zueinander, indem sie die Mengen ihrer jeweiligen Nachbarn miteinander vergleichen. Je mehr gemeinsame Nachbarn sie haben, desto dichter liegen die Knoten beieinander. In [4] konnte bereits gezeigt werden, dass dieses Vorgehen sowohl mit idealisierten, kreisförmigen als auch mit nicht idealen, unregelmäßigen Radiomodellen funktioniert. Allerdings haben wir dort lediglich den Singlehopfall untersucht, d.h. eine Distanzschätzung zwischen direkten Nachbarn. Für praktische Anwendungen ist jedoch eine Abstandsmessung über mehrere Hops hinweg in aller Regel unentbehrlich: nur so lässt sich der Abstand zu genügend vielen Ankerknoten bestimmen, so dass die Multilateration zur Positionsbestimmung zuverlässig durchgeführt werden kann.

Im vorliegenden Beitrag stellen wir daher Lösungsmöglichkeiten vor, um mit NIDES auch Entfernungen über mehrere Hops hinweg mit großer Genauigkeit ermitteln zu können. In Abschnitt 2 wird zunächst ein Überblick über die wichtigsten verwandten Arbeiten zum Thema Abstandsschätzungen in Sensornetzwerken gegeben. In Abschnitt 3 wird die grundlegende Funktionsweise von NIDES noch einmal zusammenfassend dargestellt. Die Forschungsergebnisse zur Multihop-Distanzschätzung werden in Abschnitt 4 präsentiert. In Abschnitt 5 sind schließlich die wichtigsten Resultate zusammengefasst und zudem mögliche Ansatzpunkte für weitere Forschungsarbeit in diesem Bereich aufgezeigt.

2 Verwandte Arbeiten

Im Folgenden geben wir zunächst einen Überblick über die wichtigsten in der Literatur beschriebenen Verfahren und diskutieren ihre jeweiligen Vor- und Nachteile. Wir betrachten zunächst Verfahren zur Distanzschätzung zwischen direkt benachbarten Knoten und gehen dann im Weiteren auf Techniken ein, die eine Distanzschätzung über mehrere Hops erlauben.

2.1 Distanzschätzung zwischen direkten Nachbarn

Eines der ersten Verfahren zur Ermittlung der Distanzen in Ubiquitous-Computing-Umgebungen ist die Detektion räumlicher Nähe (engl.: proximity detection). Implementierungen dieses Ansatzes sind beispielsweise das *Active Badge* [18] und das *Hybrid Indoor* [6] Navigationssystem. Beide arbeiten mit Infrarotsignalen,

welche die Knoten im Netz in periodischen Zeitabständen aussenden. Empfängt ein Gerät das Signal eines anderen, so kann es daraus schließen, dass es sich in Empfangsreichweite befindet. Diese beträgt bei der verwendeten Infrarottechnik nur wenige Meter. Nachteilig wirkt sich bei diesem Verfahren vor allem aus, dass die Knotendichte sehr hoch sein muss.

Bei Verfahren, die auf der Auswertung der *Signallaufzeit* (engl: *time of flight, ToF*) beruhen, wird zu einem Zeitpunkt t_1 ein Signal vom Sender zum Empfänger geschickt, t_1 wird im Signal in Form eines Zeitstempels mit codiert. Zum Zeitpunkt t_2 trifft das Signal beim Empfänger ein. Aus der Differenz $t_2 - t_1$ berechnet der Empfänger mit die Entfernung zum Sender. Ein besonders bekanntes Beispiel dieser Technik ist das *Global Positioning System (GPS)* [10]. Problematisch bei dieser Technik ist insbesondere in Sensornetzen die Tatsache, dass Signalsender und -empfänger sehr genau synchronisierte Uhren benötigen, um die Signallaufzeit exakt bestimmen zu können.

Die *differentielle* Messung der Signallaufzeit (engl.: Differential Time of Arrival, DTOA) umgeht diesen Nachteil. Hier werden zwei unterschiedliche Signale gleichzeitig ausgesendet, typischerweise ein Ultraschall- und ein Funksignal, welche sich mit unterschiedlichen Geschwindigkeiten ausbreiten. Aus der Laufzeitdifferenz kann der Empfänger die Entfernung zum Sender bestimmen. Anwendungsbeispiele für diese Technik sind die Systeme *Calamari* [19], *Cricket* [15] und *AHLoS* [16]. Allerdings eignet sich diese Verfahrensklasse für praktische Anwendungen nur sehr bedingt: Zum einen ergeben sich akzeptable Abschätzungen nur in einem Abstandsbereich zwischen 3 bis 15 m [16]. Zum anderen, besonders nachteilhaft in Sensornetzen, wird neben der ohnehin vorhandenen Funkschnittstelle ein zusätzliches Ultraschallsender/-empfängerpaar benötigt.

Verfahren, die auf *Signalstärkemessungen (engl.: received signal strength indicator, RSSI)* basieren, kommen dagegen völlig ohne zusätzliche Hardware aus. Sie nutzen für die Abstandsschätzung die ohnehin vorhandene Funkschnittstelle. Je größer die Entfernung ist, desto schwächer ist das empfangene Signal. Entsprechende Ansätze werden beispielsweise in [3] oder [2] vorgestellt. Negativ auf die Schätzung wirken sich vor allem die Mehrwegeausbreitung in Verbindung mit Reflexion, Brechung und Überlagerung sowie die Abschirmung von Radiowellen durch Hindernisse aus. Folglich gelten Schätzverfahren, die ausschließlich auf dem RSSI-Wert basieren, als recht ungenau [8].

Eine Verbesserung der RSSI-Technik wird in [12] vorgestellt. Die Autoren adaptieren die aus der Satellitentechnik bekannte Methode der Radiointerferometrie für den Bereich Sensornetze. Sie erreichen damit einen durchschnittlichen Schätzfehler von lediglich 3 cm bei einem maximalen Abstand von 160 m zwischen Sender und Empfänger. Allerdings ist die erreichte Genauigkeit nur dann erzielbar, wenn über einen Zeitintervall von mehreren Minuten gemessen wird - somit ist dieses Verfahren für mobile Anwendungen gänzlich ungeeignet. Zudem ist die Schätzung des Abstandes nach diesem Verfahren äußerst rechenintensiv und zudem sehr anfällig gegenüber Störgrößen, welche das Funksignal überlagern.

In jüngsten Arbeiten wurde eine ganz neue Verfahrensklasse zur Abstandsbestimmung entwickelt, welche auf dem Vergleich von Nachbarschaftslisten beruht: Haben zwei Knoten viele gemeinsame Nachbarn, so liegen sie dicht zusammen, gibt es dagegen nur wenige gemeinsame Nachbarn, liegen sie weiter auseinander. Eine erste Anwendung dieser Beobachtung wurde bereits 2003 in [7] vorgestellt. Die Autoren schlagen vor, den Anteil gemeinsamer Nachbarn für die Steuerung der Datenweiterleitung beim Fluten einzusetzen: je mehr Nachbarn der Empfänger einer Nachricht mit ihrem Sender gemein hat, mit desto geringerer Wahrscheinlichkeit leitet er die Nachricht weiter. Die Autoren geben allerdings kein Modell an, auf dessen Basis man eine Abschätzung des räumlichen Abstands in Metern vornehmen könnte. Ein solches Modell wurde von Buschmann et al. in [5] vorgestellt. Anhand von geometrischen Überlegungen zeigen die Autoren, wie mittels der Mächtigkeiten von Nachbarschaftsschnittmengen Abstände geschätzt werden können. Eine genaue Beschreibung dieses Ansatzes findet der Leser in Abschnitt 3.

2.2 Multihop-Distanzschätzung

Allen bisher betrachteten Verfahren ist gemein, dass sich hiermit lediglich Abstände zwischen Geräten schätzen lassen, die direkt miteinander kommunizieren können. Bei etlichen Sensornetzanwendungen ist es jedoch wünschenswert, dass sich auch Abstände zwischen Geräten bestimmen lassen, die nicht direkt miteinander kommunizieren können; d.h. die Entfernung zwischen den Geräten ist größer als die Kommunikationsreichweite. Auch hierzu wurden bereits Arbeiten durchgeführt.

Der einfachste Ansatz besteht darin, die Entfernungen, die zwischen zwei direkt benachbarten Knoten geschätzt werden, über die einzelnen Hops entlang eines Multihoppfades aufzuaddieren: Damit alle Geräte die Distanz zu einem Knoten ermitteln können, sendet dieser eine Flutwelle aus. Jedes Gerät addiert beim Weiterleiten der Nachricht seine Distanz zum Vorgänger auf die in der Nachricht enthaltene Distanz auf. Lernt ein Gerät auf diese Weise einen Pfad mit einer kürzeren Distanz zum ursprünglichen Absender kennen, leitet es die Nachricht weiter, sonst verwirft es sie. Ein entsprechendes Verfahren wird in [17] beschrieben. Im Folgenden werden wir diesen Ansatz mit *Sum-Dist* bezeichnen - die Autoren von [11] verwenden diese Bezeichnung ebenfalls.

Ein Nachteil von *Sum-Dist* besteht darin, dass sich die Schätzfehler über mehrere Hops akkumulieren können: Wird bei einem Verfahren stets über- oder unterschätzt, zum Beispiel weil Umwelteinflüsse die Messungen zeitweilig beeinträchtigen, summieren sich die Schätzfehler. Niculescu und Nath schlagen in [13] *DV-Hop* vor. Bei diesem Verfahren wird lediglich die minimale Anzahl der Weiterleitungen (statt der geschätzten Abstände) durch das Netzwerk propagiert. Kennt man die mittlere Hoplänge, kann man diese mit der Anzahl der erforderlichen Hops multiplizieren und erhält so eine Abstandsschätzung. Zur Ermittlung der mittleren Hoplänge ist ein separater Kalibrierungsschritt erforderlich, bei dem man davon ausgeht, dass im Netzwerk mehrere Geräte vorhanden sind,

die ihren Abstand zueinander kennen. Diese können nach Empfang der ersten Flutwelle die mittlere Hoplänge berechnen und im Netz propagieren.

Sowohl Sum-Dist und DV-Hop berücksichtigen keine geometrischen Abhängigkeiten, d.h. die geschätzten Distanzen werden nicht dahingehend überprüft, ob sie in der 2D-Ebene überhaupt geometrisch möglich sind. Diesen Aspekt greift das Verfahren *Euclidean* [13] von Niculescu und Nath auf. Hier werden geometrische Beziehungen zwischen mehreren Knoten ausgenutzt, um Distanzen so zu schätzen, dass sie in der 2D-Ebene auch möglich sind. Das Verfahren hat den Nachteil, dass nicht jeder Knoten zu einer Distanzschätzung gelangt. Eine zusammenfassende Darstellung dieses Verfahrens findet der Leser in [11].

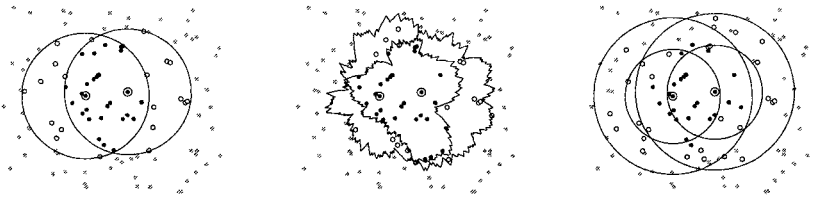
3 Überblick über die Funktionsweise von NIDES

Das *neighborhood intersection distances estimation scheme (NIDES)* wurde erstmalig in [5] vorgestellt. Es basiert auf der Idee, dass nahe bei einander liegende Knoten im Allgemeinen mehr gemeinsame Nachbarn haben als weiter von einander entfernte. Nimmt man an, dass Knoten mit Nachbarn innerhalb einer gewissen Reichweite r kommunizieren können, ergibt sich als Kommunikationsbereich ein Kreis. Auf Basis dieses Kreismodells kann man mittels geometrischer Überlegungen einen funktionalen Zusammenhang zwischen dem Anteil der Kreisüberdeckung und dem Abstand der Kreismittelpunkte herstellen. Nimmt man eine Gleichverteilung der Knoten an, kann man so vom Anteil der gemeinsamen Nachbarn auf die Distanz der betreffenden Knoten schließen. Details zum Modell können [5] entnommen werden.

In drahtlosen Netzwerken ist die Funkausbreitung jedoch in den seltensten Fällen wirklich kreisförmig. Dies gilt insbesondere in Sensornetzwerken mit ihren einfachen Funkschnittstellen [9]. Vielmehr ist die Kommunikationsreichweite eines Knotens richtungsabhängig und zeitlich variabel. Diese Erkenntnisse wurden in komplexeren Funkausbreitungsmodellen wie dem RI-Modell oder dem stochastischen Modell aufgenommen. Das *radio irregularity model* [20] beschreibt die Kommunikationsreichweite als Funktion des Winkels, so dass sich statt eines Kreises eine unregelmäßige Form ergibt. Der Parameter *degree of irregularity (DOI)* gibt dabei den Grad der Unregelmäßigkeit der resultierenden Form an. Beim *stochastischen Modell* [21] ist Kommunikation zwischen Knoten, deren Distanz oberhalb einer Schranke liegt, nur mit einer gewissen Wahrscheinlichkeit möglich. Diese nimmt mit zunehmender Distanz ab.

In Abbildung 1 sind die Nachbarschaften zweier Knoten für die drei genannten Funkausbreitungsmodelle gezeigt. Offensichtlich variiert die Anzahl der gemeinsamen Nachbarn bei ansonsten unveränderten Bedingungen zwischen 26 (Kreismodell, Abbildung 1(a)), 27 (RI-Modell, Abbildung 1(b)) und 23 (stochastisches Modell, Abbildung 1(c)).

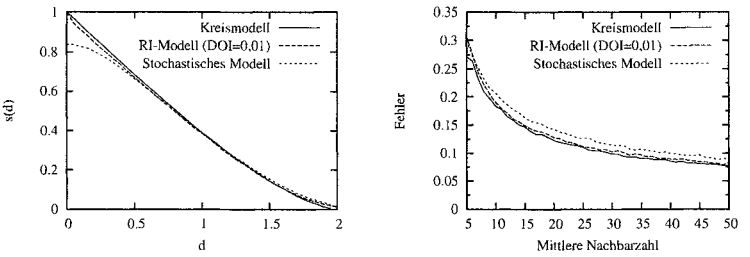
Daher wurde das Verfahren zur Distanzschätzung, welches ursprünglich nur für das Kreismodell anwendbar war, in [4] so verallgemeinert, dass es auch für andere Funkausbreitungsmodelle wie das RI-Modell oder das stochastische Modell anwendbar ist. Dabei wird aus dem Ausbreitungsmodell der Verlauf der



(a) Kreismodell. (b) RI-Modell (DOI=0.01). (c) Stochastisches Modell.

Abb. 1. Die Anzahl der gemeinsamen Nachbarn zweier Knoten in Abhängigkeit vom Radiomodell.

Schätzfunktion $s(d)$ bestimmt, die dann den Zusammenhang zwischen dem Anteil der gemeinsamen Nachbarn und der Distanz zwischen den Knoten darstellt.



(a) Schätzfunktion $s(d)$. (b) Schätzfehler in Abhängigkeit von der mittleren Anzahl von Nachbarn.

Abb. 2. Schätzfunktion und Schätzfehler für verschiedene Funkausbreitungsmodelle.

Abbildung 2(a) zeigt exemplarisch für das RI-Modell den Verlauf der Schätzfunktion, d.h. den erwarteten Anteil gemeinsamer Nachbarn $s(d)$ in Abhängigkeit von der Distanz d. Dabei ist die Distanz auf den mittleren Kommunikationsradius r normiert. Mit Hilfe dieser Funktion können dann Distanzen geschätzt werden.

Das Protokoll, das hier beschriebene Verfahren umsetzt, wurde in [5], [4] im Detail beschrieben. Es benötigt lediglich 2 versendete Nachrichten pro Gerät, weniger als 50 Byte Speicher und ist auch auf einfachsten Controllern implementierbar.

Die Genauigkeit der Schätzungen wurde mit Hilfe von Simulationen mit ns-2 [14] überprüft. Abbildung 2(b) zeigt den Schätzfehler für verschiedene Ausbreitungsmodelle in Abhängigkeit von der Netzdichte, d.h. der mittleren Anzahl von Nachbarn. Dabei ist der Fehler als Bruchteil des Kommunikationsradius

r angegeben. Es ist zu erkennen, dass es kaum Unterschiede zwischen den verschiedenen Funkausbreitungsmodellen gibt. Der mittlere Schätzfehler nimmt mit zunehmender Netzdichte kontinuierlich ab. Er liegt bei etwa 20% des mittleren Kommunikationsradius bei einer Dichte von 10 und fällt auf 8% des mittleren Kommunikationsradius bei einer Dichte von 50 ab. Das bedeutet, dass die Schätzungen mit zunehmender Dichte immer genauer werden. Die Schätzungen sind ausgewogen, dass heißt es gibt keine Tendenz zu systematischen Über- oder Unterschätzungen. Details können [5], [4] entnommen werden. Dort wurde auch gezeigt, dass NIDES auch in Netzwerken mit variabler Netzdichte gleichbleibende Leistungen erzielt.

4 Multihop-Distanzschätzung: NIDES in Verbindung mit Sum-Dist

4.1 Simulationsumgebung

Um die Genauigkeit von Multihop-Distanzschätzungen auf Basis von Nachbarschaften zu untersuchen, haben wir eine Reihe von Simulationen mit ns-2 [14] durchgeführt. Dazu war er erforderlich, NIDES so zu erweitern, dass Distanzen über mehrere Hops hinweg akkumuliert werden können. Wir haben uns für das Sum-Dist-Verfahren (vergleiche Abschnitt 2.2) entschieden, da es schnell, umfassend und ohne weitere Voraussetzungen funktioniert. DV-Hop hingegen benötigt Referenzdistanzen zur Bestimmung der mittleren Hoplänge, das Eucledian-Verfahren arbeitet zwar genauer, allerdings kommen Knoten in verschiedensten Fällen gar nicht zu einer Distanzschätzung.

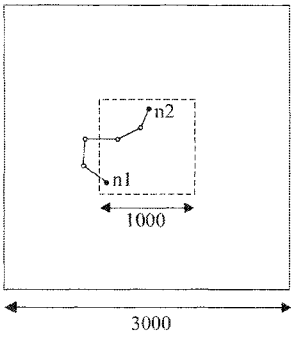


Abb. 3. Simulationsszenario.

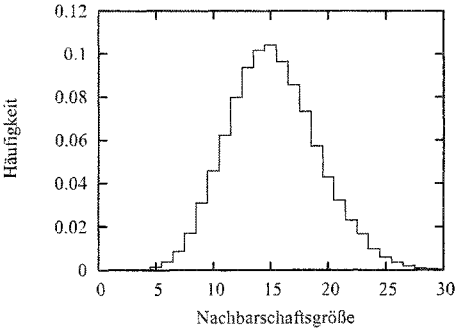


Abb. 4. Histogramm der Nachbarschaftsgrößen

Für die Simulationen wurde das RI-Model implementiert, indem das Funkausbreitungsmodell des Simulators erweitert wurde. In der 3000 x 3000 Meter großen Simulationsfläche haben wir die Knoten gleichverteilt angeordnet (Abbildung 3). Die mittlere Kommunikationsreichweite betrug 100 Meter. Die Anzahl

der Knoten haben wir so festgelegt, dass sich im Mittel die gewünschte Netzwerkdichte ergab. Dann wurden die Distanzen zwischen benachbarten Knoten mit Hilfe von NIDES geschätzt. Im Anschluss wurden für alle Knotenpaare die kürzesten Pfade untereinander ermittelt, wie es das Sum-Dist-Verfahren vorsieht. Entlang dieser Pfade wurden dann die Distanzschätzungen der einzelnen Hops aufaddiert und mit der euklidischen Distanz der Knoten an den Pfadenden verglichen.

Für die Bewertung der Distanzschätzungen wurden nur Pfade zwischen Knotenpaaren herangezogen, die im zentral liegenden Neuntel der Simulationsfläche (gestrichelt) lagen. Auf diese Weise können Randeffekte ausgeblendet werden. So könnte es bei Knoten am Rand passieren, dass außerhalb der Simulationsfläche ein günstigerer Pfad hätte gefunden werden können. Durch die verwendete Pufferzone kann dieses weitgehend ausgeschlossen werden (wie bei dem Pfad zwischen n1 und n2).

Um statistisch verlässliche Aussagen zu erhalten, haben wir jeweils die Ergebnisse von 100 Simulationsläufen gemittelt.

4.2 Eigenschaften von Multihop-Adhoc-Netzwerken

Die Analyse von Szenarien mit zufällig gleichverteilten Knoten zeigt, dass die Netzwerkdichte keineswegs gleichmäßig ist. Vielmehr ergibt sich eine Verteilung der Nachbarschaftsgrößen, die einer Normalverteilung um die eingestellte Netzdichte gleicht. Abbildung 4 zeigt exemplarisch ein solches Histogramm der Nachbarschaftsgrößen.

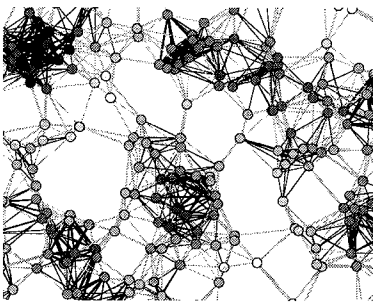


Abb. 5. Ausschnitt aus einem Szenario.

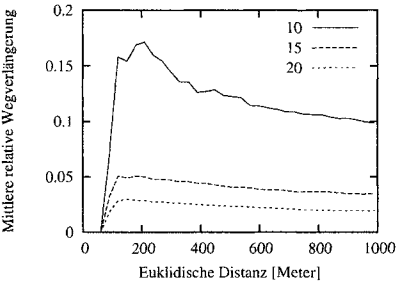


Abb. 6. Wegverlängerung bei Multihop-pfaden

In Abbildung 5 ist ein Ausschnitt aus einem Szenario der Dichte 15 zu sehen. Sie zeigt Knoten sowie die sich ergebenden Kommunikationsverbindungen zwischen ihnen. Der Grauton der Knoten repräsentiert dabei die Anzahl der Nachbarn: weiß steht für wenige, schwarz für viele Nachbarn, Grauschattierungen stehen für mittlere Dichten. Es ist klar zu erkennen, dass sich Cluster höherer

Dichte und dazwischen Bereiche mit weniger Knoten bilden, die im Extremfall zu Löchern in der Topologie entarten können.

Solche Löcher führen dazu, dass die Multihoppfade nicht gerade, sondern in Abhängigkeit von der Netzdichte mehr oder weniger stark im Zickzack verlaufen. Aufgrund dieser Tatsache ist die Summe der Distanzen zwischen den benachbarten Knoten im Pfad stets größer als die euklidische Distanz der Knoten an beiden Enden des Pfades. Diese Verlängerung des Weges ist von der Netzdichte abhängig: Je geringer die Dichte, desto länger sind die Pfade im Vergleich zur euklidischen Distanz. Abbildung 6 zeigt diesen Effekt beispielhaft für die Dichten 10, 15 und 20. Es ist weiterhin erkennbar, dass dieser Effekt bei kurzen Distanzen ausgeprägter ist als bei langen. Ein Grund dafür ist, dass bei großen Distanzen mehr Hops zwischen den Knoten liegen und es somit mehr Möglichkeiten für günstige Pfade gibt. Weiterhin werden für Knoten mit geringerer Distanz zum Teil Pfade gefunden, die sehr viel länger sind als die euklidische Distanz, weil sie große Umwege nehmen. Für Knoten mit großer Distanz finden sich derartig lange Umwege nicht, findet sich kein kurzer Pfad, liegen sie in unterschiedlichen Netzpartitionen.

Einzelne Links durchqueren die Löcher zwischen Clustern und verbinden so die Knoten auf der einen Seite mit denen auf der anderen. Wir wollen sie im Weiteren *Brückenlinks* nennen. Ihre verbindende Eigenschaft führt dazu, dass sie mit überdurchschnittlicher Wahrscheinlichkeit in Multihoppfaden durch das Netzwerk enthalten sind. Das hat zur Folge, dass sich Fehler beim Schätzen von Distanzen entlang von Brückenlinks überproportional auf die Multihopschätzung niederschlagen.

In Abbildung 5 sind solche Links schwarz eingetragen, entlang derer es bei Distanzschätzungen mit NIDES zu Unterschätzungen kam, grau steht für einen Link, entlang dem überschätzt wurde. Die Dicke der Linie steht für den Grad der Über- bzw. Unterschätzung: eine dicke Linie repräsentiert eine starke Verschätzung. Es fällt auf, dass die häufig verwendeten Brückenlinks fast immer überschätzt werden, und das in einem zum Teil erheblichen Maße. Die Ursache dafür ist, dass die Knoten an beiden Seiten des Links meist keine weiteren gemeinsamen Nachbarn haben. Daher gelangen sie zu einer Schätzung, die etwa beim Zweifachen ihrer mittleren Kommunikationsreichweite r liegt (vergleiche Abbildung 2(a)), obwohl sie auch beim RI-Modell im Mittel nicht weiter als r von einander entfernt sind.

4.3 Anpassung an Multihop-Distanzschätzung

Um den im vorigen Abschnitt beschriebenen Ausreißern entgegen zu wirken schlagen wir vor, NIDES leicht zu modifizieren. Geschätzte Distanzen, die oberhalb von r liegen, sind im Allgemeinen das Resultat von Überschätzungen, da es sehr unwahrscheinlich ist, dass zwei Knoten mit einem Abstand von mehr als r durch einen Link verbunden sind. Daher ist es sinnvoll, die Schätzungen auf den mittleren Kommunikationsradius zu begrenzen. Die entsprechend angepassten Schätzfunktionen (aus Abbildung 2) sind in Abbildung 7 dargestellt. Diese

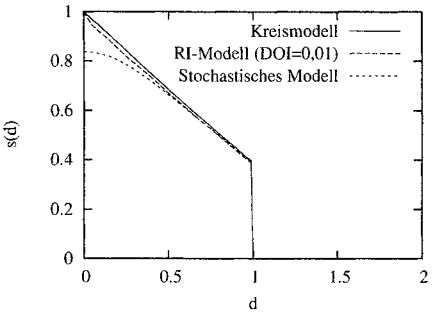


Abb. 7. Modifizierte Schätzfunktionen.

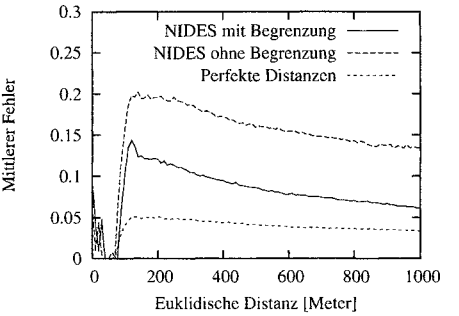


Abb. 8. Prozentualer Schätzfehler

Limitierung wirkt Überschätzungen effektiv entgegen, ohne das Schätzverfahren ansonsten zu beeinträchtigen.

4.4 Ergebnisse

Im Rahmen der simulativen Evaluation wurde untersucht, wie sehr die Multihop-Distanzschätzungen von der euklidischen Distanz der Knoten an den beiden Enden des Multihoppfades abweicht. Dabei wurde das RIM-Funkausbreitungsmodell verwendet. Die Netzwerkdichte betrug 15, das heißt jeder Knoten hatte im Mittel 15 Nachbarn.

Um den Erfolg der Modifikation von NIDES einschätzen zu können, wurden sowohl Simulationen mit der modifizierten Version von NIDES mit Begrenzung der Schätzungen auf den mittleren Kommunikationsradius wie auch ohne durchgeführt. Als Referenz wurden darüber hinaus Simulationen mit einer *perfekten* Distanzmessung entlang der Multihoppfade durchgeführt. Diese drei Messverfahren wurden dann jeweils mit der euklidischen Distanz verglichen.

Abbildung 8 zeigt die Ergebnisse. Auf der Abszisse ist die euklidische Distanz der betrachteten Knoten aufgetragen. Die Ordinate zeigt die jeweiligen mittleren Messfehler der drei Verfahren als Bruchteil der tatsächlichen Distanz.

Es ist zu erkennen, dass die Messfehler ab einer Distanz von etwa 100 stark ansteigen. Das liegt daran, dass ab hier ein zweiter Hop zur Überwindung der Distanz erforderlich ist. Die Kurve der perfekten Distanzmessung zeigt, dass hier auch ohne Fehler in der eigentlichen Distanzschätzung circa 5% zu lange Distanzen resultieren. Die Ursache hierfür ist, wie im vorigen Abschnitt erläutert, dass die Distanz entlang des Multihoppfades tatsächlich länger ist als die euklidische, da der Pfad nicht gerade verläuft.

Der mittlere Fehler für NIDES mit Begrenzung steigt auf etwa 13% an, um dann kontinuiertlich auf 6% für große Distanzen zurückzugehen. Am Verlauf der Kurve der perfekten Distanzen wird erkennbar, dass ein Drittel bis die Hälfte dieses Fehlers auf die Eigenschaften der Multihoppfade zurückgehen und selbst mit fehlerfreier Distanzschätzung nicht vermeidbar ist.

Ebenfalls wird deutlich, dass der Fehler für NIDES ohne Begrenzung auf 20% ansteigt, und dann sukzessive auf 13% abfällt. Er liegt damit etwa 7% höher als der der Variante mit Begrenzung. Dies macht deutlich, dass die Begrenzung der Distanzschätzungen auf den mittleren Kommunikationsradius ein äußert effektives Mittel zur Fehlerreduzierung ist.

Für Distanzen unterhalb von 100 Metern (d.h. im Singlehopfall) liegt der Fehler für beide Varianten von NIDES bei etwa 5%.

5 Zusammenfassung und Ausblick

In dieser Arbeit stellten wir ein Verfahren zur Multihop-Distanzschätzung in drahtlosen Funknetzwerken vor. Es basiert auf dem Vergleich von Nachbarschaftslisten und hat den Vorteil, dass für die Distanzschätzung lediglich die ohnehin vorhandene Funkschnittstelle benötigt wird, die Ungenauigkeiten von RSSI-basierten Verfahren aber nicht auftreten.

Es wurde aufgezeigt, wie die verschiedenen Einflussfaktoren, die sich aus den Eigenschaften von zufällig gleichverteilten Adhoc-Netzen ergeben, bei der Multihop-Distanzschätzung berücksichtigt werden können. Die Genauigkeit des Verfahrens wurde mit Hilfe von Simulationen untersucht. Dabei konnte gezeigt werden, dass der mittlere Fehler der Schätzungen 13% der euklidischen Distanz nicht überschreitet, für große Distanzen liegt er bei 6%.

Wir bereiten zur Zeit Experimente mit einer großen Zahl realer Sensorknoten vor, um die vorliegenden Simulationsergebnisse experimentell zu überprüfen. Wir planen darüber hinaus, NIDES dahingehend zu erweitern, dass 2-Hopnachbarschaften betrachtet werden. Dies ermöglicht es, dass Knoten den zwischen zwei Nachbarn und sich selbst eingeschlossenen Winkel abschätzen können. Auf diese Weise können unter anderem die Winkel zwischen einzelnen Segmenten eines Multihoppfades berücksichtigt werden, um die Präzision der Distanzschätzung weiter zu verbessern.

Literaturverzeichnis

1. I. F. Akyildiz, Y. S. W. Su, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, Mar. 2002.
2. P. Bergamo and G. Mazzini. Localization in sensor networks with fading and mobility. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2002.
3. N. Bulusu, J. Heideman, and D. Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications*, 2000.
4. C. Buschmann, H. Hellbrück, S. Fischer, A. Kröller, and S. Fekete. Radio propagation-aware distance estimation based on neighborhood comparison. In *Proceedings of the 14th European conference on Wireless Sensor Networks (EWSN 2007), Delft, The Netherlands*, Jan. 2007.
5. C. Buschmann, D. Pfisterer, and S. Fischer. Estimating distances using neighborhood intersection. In *11th IEEE International Conference on Emerging Technologies and Factory Automation, Prague, Czech Republic*, Sept. 2006.

6. A. Butz, J. Baus, A. Krüger, and M. Lohse. A hybrid indoor navigation system. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 25–32, New York, NY, USA, 2001. ACM Press.
7. J. Cartigny and D. Simplot. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, page 303, Washington, DC, USA, 2003. IEEE Computer Society.
8. E. Elnahrawy, X. Li, and R. P. Martin. The limits of localization using signal strength: A comparative study. In *EEE SECON*, 2004.
9. D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks, 2002.
10. B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer, 5 edition, 2001.
11. K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Networks*, 2003.
12. M. Maroti, P. Völgyesi, S. Dora, B. Kusy, A. Nadas, A. Ledeczi, G. Balogh, and K. Molnar. Radio interferometric geolocation. In *SenSys "05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, 2005.
13. D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *Proceedings of GLOBECOM, San Antonio, November 2001.*, 2001.
14. The Network Simulator ns-2 (v2.29). http://www.isi.edu/nsnam/ns/, October 2001.
15. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, New York, NY, USA, 2000. ACM Press.
16. A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM Press.
17. A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *First ACM International Workshop on Wireless Sensor Networks and Application*, Atlanta, GA, USA, 2002.
18. R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.
19. K. Whitehouse and D. Culler. Calibration as parameter estimation in sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 59–67, New York, NY, USA, 2002. ACM Press.
20. G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 125–138, New York, NY, USA, 2004. ACM Press.
21. M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)*, October 2004.

Verhaltensbeobachtung und -bewertung zur Vertrauensbildung in offenen Ad-hoc-Netzen

Daniel Kraft[1] und Günter Schäfer[2]

[1] Institut für Telematik, Universität Karlsruhe
[2] Fachgebiet Telematik/Rechnernetze, Technische Universität Ilmenau

Zusammenfassung. Die Beobachtung des Verhaltens anderer Teilnehmer anhand mitgehörten Netzverkehrs ist in offenen Ad-hoc-Netzen die einzige ohne Vorwissen anwendbare Methode, durch welche Einschätzungen über die Kooperationsbereitschaft und Vertrauenswürdigkeit Anderer gewonnen werden können. Dieser Beitrag beschreibt die Problematik der Verhaltensbeobachtung und entwickelt auf der Basis einer grundlegenden Bedrohungsanalyse ein konkretes Verfahren für die zuverlässige Bewertung des Weiterleitungsverhaltens teilnehmender Knoten in Ad-hoc-Netzwerken. Anhand von Simulationsergebnissen wird nachgewiesen, dass mit dem Verfahren eine ausreichende Anzahl an Bewertungsereignissen gewonnen werden kann.

1 Einleitung

Vertrauen in andere Teilnehmer ist eine grundlegende Voraussetzung für die Funktionsfähigkeit jedes verteilten Systems, in dem Dienstleistungen erbracht und in Anspruch genommen werden. Bei existierenden verteilten Systemen wird häufig einfach generell in korrektes Verhalten aller Teilnehmer vertraut – eine Annahme, die bei großen offenen Systemen ggf. nicht mehr gerechtfertigt ist, wenn die Nutzer die Systeme lediglich als Werkzeug für die Verfolgung ihrer eigentlichen Interessen sehen.

In offenen Ad-hoc-Netzen gibt es im Wesentlichen zwei Möglichkeiten, Vertrauen in andere Teilnehmer zu gewinnen: Entweder muss existierendes Vertrauen aus der realen Welt in die virtuelle des Netzwerks übertragen werden (was neben der Existenz solchen Vertrauens eine sichere Verknüpfung zwischen realen Benutzern und Identitäten im Netz agierender Rechner voraussetzt und aktive Mitarbeit des Benutzers erfordert) oder neues Vertrauen muss innerhalb der Netzwerkwelt aufgebaut werden, indem das Verhalten der anderen Knoten automatisch beobachtet und bewertet wird und somit Erfahrungen gesammelt werden, ähnlich wie dies Menschen beim Aufbau sozialer Beziehungen und Netzwerke tun. Benachbarte Netzknoten können hierbei dank der Rundrufcharakteristik drahtloser Übertragungstechniken anhand mitgehörten Netzverkehrs beobachtet werden. Da die Möglichkeiten automatischer Beobachtung in der virtuellen Welt gegenüber der realen damit allerdings recht eingeschränkt sind (im Allgemeinen kann nur die Einhaltung von Protokollen und die Reaktion auf Dienstleistungsanforderungen automatisch beurteilt werden), ist es besonders wichtig, möglichst viel der zur Verfügung stehenden Information zu nutzen. Insbesondere sollte sowohl *beobachtetes korrektes Verhalten* als auch *beobachtetes inkorrektes Verhalten* explizit registriert

werden und in die ermittelte Einschätzung einfließen: Korrektes Verhalten erzeugt Vertrauen beim Beobachter, inkorrektes Misstrauen. Liegen gar keine Beobachtungen vor, so ist weder Vertrauen noch Misstrauen angebracht. Die verwendete Vertrauensmetrik muss geeignet sein, dies auszudrücken.

Die Beobachtung des Verhaltens anderer Knoten in Bezug auf die Erbringung der Paketweiterleitung in der Netzwerkschicht erscheint aufgrund der häufigen Nutzung dieses Dienstes als besonders geeignet für die Einschätzung der Kooperationsbereitschaft anderer Knoten. Die Verhaltensbeobachtung enthält hierbei jedoch grundsätzlich gewisse Unsicherheiten. Beispielsweise kann sich der zur Weiterleitung verpflichtete Knoten zwischen Empfang und Weiterleitung eines Pakets aus der Reichweite des Beobachters bewegen oder auf andere Weise von ihm abgeschirmt werden, so dass der Beobachter die Weiterleitung nicht beobachten kann und fälschlich annimmt, dass sie unterlassen wurde. Sind die Bedingungen für eine richtige Bewertung aber im Großteil der Fälle erfüllt und führen einzelne Beobachtungen immer nur zu graduellen Anpassungen von Einschätzungen, können wenige „falsche" Beobachtungen i. d. R. ohne wesentliche Verfälschungen der Gesamteinschätzungen verkraftet werden.

Der vorliegende Beitrag stellt ein Verfahren zur Beobachtung und Bewertung des Weiterleitungsverhaltens in Ad-hoc-Netzen vor, das nicht von einem bestimmten Routing-Protokoll abhängig ist und alle beobachtbaren Übertragungen einbezieht. Der folgende Abschnitt beschreibt zunächst verwandte Ansätze. Abschnitt 3 beschreibt unseren Ansatz, dessen wesentliche Neuerungen anhand einer systematischen Bedrohungsanalyse begründet werden. Die Leistungsfähigkeit des Verfahrens wird in Abschnitt 4 in einer Simulationsstudie untersucht. Abschnitt 5 fasst die Ergebnisse des Beitrags kurz zusammen und gibt einen Ausblick auf weiterführende Arbeiten.

2 Stand der Technik

Marti, Giuli, Lai und Baker beschreiben [MGLM00] den ersten Ansatz zur Beobachtung des Weiterleitungsverhaltens in Ad-hoc-Netzen, der speziell für das reaktive Routing-Verfahren Dynamic Source Routing (DSR)entworfen wurde. Dabei bewahrt eine „Watchdog" genannte Komponente eine Kopie eines jeden in der Weiterleitungsphase selbst gesendeten Pakets auf, das noch der Weiterleitung bedarf, und startet mit der Aussendung einen Zeitgeber. Wird in der Folge beobachtet, wie das Paket weitergeleitet wird, so werden Kopie und Zeitgeber gelöscht. Läuft der Zeitgeber aber ab, ohne dass die Weiterleitung beobachtet wurde, so erfolgt eine Meldung über unterlassene Weiterleitung an die Quelle des Pakets; sie führt zum Ausschluss von Wegen über den unzuverlässigen Knoten bei der Wegewahl. In der Routing-Phase selbst erfolgt keine Überwachung. Durch die Beschränkung auf ein Source-Routing-Protokoll kann der Beobachter leicht erkennen, ob und wohin ein Paket noch weitergeleitet werden muss.

Dieser Ansatz wurde häufig in ähnlicher Form aufgegriffen. Beim CONFIDANT-Ansatz [BuBo02] etwa wird ein ähnliches Beobachtungsverfahren verwendet, um negative Bewertungen zu ermitteln, die bei gehäuftem Auftreten zum Ausschluss des Verursachers aus dem Netz führen. In einer Weiterentwicklung [BuBo04,BuTLB04] wird auch korrektes Verhalten registriert und aus positiven und negativen Beobachtungen eine Einschätzung ermittelt.

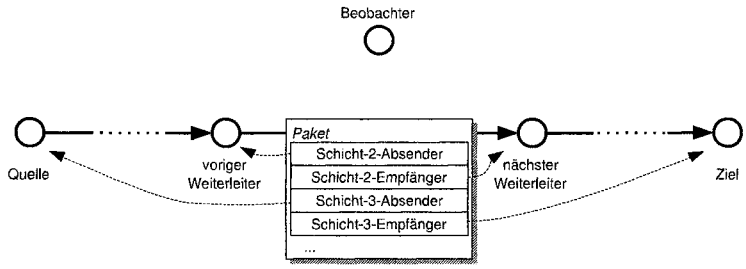


Abb. 1. An der Übertragung eines Pakets beteiligte Knoten und Adressangaben in Paketen

Auch bei SORI [HeWK04] und OCEAN [BaBa03] gibt es jeweils eine Watchdog-Komponente, welche positive und negative Beobachtungen bezüglich der selbst versendeten Pakete erfassen. Pakete, die von anderen Teilnehmern versendet worden sind, werden aber auch hier nicht beobachtet. In [BaBa03] wird dies dadurch begründet, dass solche Beobachtungen zu anfällig gegen Angriffe seien.

3 Beobachtung des Weiterleitungsverhaltens

Zur Klärung der Begriffe in den weiteren Ausführungen sind in Abbildung 1 einige an der Übertragung eines Pakets beteiligte Knoten dargestellt und aus der Sicht eines Beobachters bezeichnet. Ein allgemeiner Grundsatz für die Verfahren zur Beobachtung und Bewertung ist, dass durch sie möglichst wenig zusätzlicher Energieaufwand und keine zusätzliche Netzbelastung entstehen sollten.

3.1 Grundidee

Jeder Knoten hält für jeden seiner Nachbarn ein Paar von Zählern, mit denen er Beobachtungen korrekten bzw. unkorrekten Weiterleitungsverhaltens registriert. Aus den Zählerständen kann jederzeit eine Vertrauensmaßzahl berechnet werden, die eine Einschätzung des Weiterleitungsverhaltens des jeweiligen Nachbarn darstellt.

Beobachtet ein Knoten, wie ein anderer Knoten ein Paket weiterleitet, erhöht er den diesem weiterleitenden Knoten zugeordneten Zähler für positive Beobachtungen um eins. Um auch erfassen zu können, dass ein Knoten die Weiterleitung verweigert, wird außerdem für jedes beobachtete oder selbst ausgesandte Paket, das noch von einem Nachbarn weitergeleitet werden muss, ein Eintrag in einer Liste derzeit beobachteter Pakete angelegt. Zu jedem solchen Eintrag wird ein Zeitgeber gestartet. Wird während der Laufzeit des Zeitgebers die erwartete Weiterleitung beobachtet, so werden der Zeitgeber und der zugehörige Eintrag gelöscht. Läuft der Zeitgeber aber ab, ohne dass die erwartete Weiterleitung beobachtet werden konnte, so wird dem säumigen Weiterleiter eine negative Beobachtung angerechnet.

Führen die einzelnen Knoten eine Zugangskontrolle in dem Sinne durch, dass Pakete nur dann weitergeleitet werden, wenn sie von einem vertrauenswürdigen Knoten stammen [Kraf06], so ist für Beobachter zunächst nicht vorhersehbar oder erkennbar,

wie die aufgrund lokaler Information getroffenen Zugangsentscheidungen ausfallen. In Fällen unterlassener Weiterleitung kann somit nicht entschieden werden, ob die Weiterleitung berechtigterweise aufgrund einer Zugangsentscheidung oder unberechtigterweise unterlassen wurde. Daher muss jeder Knoten, der eine negative Zugangsentscheidung trifft, dies durch Aussenden einer entsprechenden Fehlermeldung bekanntgeben. Wenn eine solche Fehlermeldung beim Beobachter eintrifft, wird das zugehörige Paket aus der Liste beobachteter Pakete gelöscht.

Als Teilnehmerkennungen, denen positive und negative Beobachtungen zugeordnet werden, dienen die öffentlichen Teile asymmetrischer Schlüsselpaare [Kraf06]. Ein Teilnehmer kann und darf sich mehrere solcher Kennungen zulegen. Aus dem Vorliegen unterschiedlicher Kennungen etwa als Absender zweier Nachrichten kann somit nicht geschlossenen werden, dass diese von unterschiedlichen Teilnehmern stammen. Andererseits kann aber ausschließlich der tatsächliche Inhaber einer Teilnehmerkennung Signaturen mit dem privaten Teil des asymmetrischen Schlüsselpaars erzeugen, die dann mit Hilfe des öffentlichen Teils verifizierbar sind. Fremde Teilnehmerkennungen können somit nicht für eigene Zwecke missbraucht werden. Die Teilnehmerkennungen seiner Nachbarn sind jedem Knoten stets bekannt, da jeweils beim ersten Kontakt mit neuen Nachbarn ein Schlüsselaustausch durchgeführt wird, bei welchem jede Seite auch den Besitz des privaten Schlüssels nachweist (Challenge-Response-Verfahren).

Bei dem beschriebenen Beobachtungs- und Bewertungsverfahren werden an zwei Stellen Informationen über die Zuordnung zwischen Schicht-2- und Schicht-3-Adressen benötigt, die nicht aus dem beobachteten Paket entnommen werden können und daher durch eine entsprechende Abbildungsfunktion (bzw. Routing-Information) ermittelt werden müssen. Gemeint sind die beiden folgenden Entscheidungen:

- Wenn ein Sendevorgang beobachtet wird, muss zunächst entschieden werden, ob es sich um einen Weiterleitungsvorgang handelt, oder ob die beobachtete Nachricht von ihrem ursprünglichen Erzeuger (also ihrer Quelle) ausgesandt wurde.
- Weiterhin muss ein Beobachter erkennen, ob ein beobachtetes Paket nochmals weitergeleitet werden muss, oder ob es am Ziel angekommen ist.

3.2 Bedrohungsanalyse

Angriffsziele und -motivation Bei dem Versuch, die Resultate des Verfahrens zur Verhaltensbeobachtung und -bewertung zu verfälschen, kommen für Angreifer folgende Ziele in Frage:

1. Beschaffung positiver Bewertungen für den Angreifer bzw. für „Komplizen", wobei
   (a) der für reguläre positive Bewertung zu erbringende Aufwand (Dienstleistung an der Allgemeinheit) entfällt oder
   (b) die sich durch mangelnde Nachfrage nach zu erbringenden Dienstleistungen ergebende Einschränkung (etwa dadurch, dass keine Pakete zur Weiterleitung zur Verfügung stehen) wegfällt.

   Eine Art natürlicher Beschränkung für die Beschaffung ungerechtfertigter positiver Bewertungen der erstgenannten Art besteht dadurch, dass immer nur bei Beobachtung eines tatsächlich gesendeten Pakets eine positive Bewertung erfolgt. Dass positive Bewertungen völlig ohne Aufwand (d. h. ohne Energieverbrauch durch Senden) massenhaft erzeugt werden können, ist damit schon ausgeschlossen. Es kann

also höchstens erreicht werden, dass ein normalerweise nicht positiv bewerteter Sendevorgang (wie die Erzeugung eines eigenen Pakets) als Dienstleistung an der Allgemeinheit (z. B. Weiterleitung) erscheint.
2. Verhindern negativer Bewertung des Angreifers oder eines „Komplizen" (darf nicht mehr Aufwand erfordern, als durch unterlassene Weiterleitung eingespart wird).
3. Erzeugung negativer Bewertungen oder Verhindern positiver Bewertung für andere Knoten, entweder
   (a) um die Verfügbarkeit des Netzes zu beeinträchtigen oder
   (b) um eine eigene schlechte Bewertung durch andere Knoten zu relativieren.
   Angriffe gegen die Verfügbarkeit sind in Ad-hoc-Netzen schwierig zu bekämpfen, weil mit genügend hohem Aufwand ohnehin jede Kommunikation zumindest in der Nachbarschaft des Angreifers durch Störsignale unterbunden werden kann. Ob der Angriffsversuch (b) Erfolg haben kann, hängt vom Zugangskontrollverfahren ab.

Die unter 1. und 2. genannten Angriffsmotivationen dürften die größte Gruppe potentiell interessierter Angreifer anziehen, da Möglichkeiten, sich selbst Vorteile zu verschaffen, ohne dabei durch allzu offensichtliche Benachteiligung anderer auffällig zu werden, erfahrungsgemäß gerne genutzt werden.

Angriffsanalyse Die prinzipiellen Möglichkeiten eines Angreifers umfassen die *Unterschlagung, Zerstörung* oder *Verfälschung bewertungsrelevanter Information* in den Nachrichten anderer, die *Wiedereinspielung fremder Nachrichten* sowie die *Angabe falscher bewertungsrelevanter Information* in eigenen Nachrichten. Einige dieser Möglichkeiten werden durch das vorgeschlagene Verfahren von vornherein ausgeschlossen:

- Da nur Nachrichten von Nachbarn für die Beobachtung herangezogen werden, entfällt die Unterschlagung (die nur bei erforderlicher Weiterleitung möglich wäre).
- Die gezielte Zerstörung oder Verfälschung bestimmter bewertungsrelevanter Informationen innerhalb fremder Nachrichten ist technisch aufgrund der Eigenschaften der Signalausbreitung nur in den seltensten Fällen realisierbar. Eine Mindestvoraussetzung hierfür ist, dass sich der Angreifer in einer geeigneten Position zwischen Sender und Empfänger befindet. Normalerweise gibt es aber mehrere benachbarte Knoten, die ebenfalls Beobachtungen durchführen, und der Angreifer kann eine Übertragung nicht für alle Beobachter in gleicher Weise manipulieren.
- Die Falschangabe bewertungsrelevanter Information ist dann sinnlos, wenn der einzige Zweck der unversehrten Information darin liegt, dem Sender eine positive Bewertung zuzuordnen.

Es bleiben die Möglichkeiten der Zerstörung fremder Nachrichten, der Wiedereinspielung fremder Nachrichten und der Falschangabe bewertungsrelevanter Information in eigenen Nachrichten (ggf. um eine Nachricht so erscheinen zu lassen, als sei sie von einem anderen Knoten erzeugt worden):

1. *Fälschung der Identität des Weiterleiters:* Für positive Bewertungen kann die Identität des zu bewertenden Teilnehmers an der (ungesicherten) Schicht-2-Absenderadresse des beobachteten Pakets abgelesen werden. Durch Falschangabe könnte der

Absender nur sich selbst schaden. Für negative Bewertung unterlassener Weiterleitung muss die Identität des zu Bewertenden aus der Schicht-2-Zieladdresse des weiterzuleitenden Pakets ermittelt werden. Diese Angabe könnte also nur vom vorigen Weiterleiter gefälscht werden (abgesehen von technisch schwierig zu realisierender Überlagerung der Übertragung durch den Empfänger, die allenfalls einen Teil der Beobachter täuschen könnte). Die Fälschungsmotivation für diesen ist gering, da er keinen direkten eigenen Vorteil hätte, sondern höchstens versuchen könnte, die Einschätzungen eines Anderen in den Augen Dritter zu verschlechtern. Dazu könnte der Angreifer absichtlich Schicht-2-Empfänger angeben, die die jeweiligen Pakete gar nicht empfangen und deshalb auch nicht weiterleiten können.

2. *Erfinden angeblich weiterzuleitender Pakete:* Erzeugt ein „Weiterleiter" ein angeblich weiterzuleitendes Paket selbst oder wiederholt er ein früher bereits übertragenes Paket noch einmal, erbringt er keine positiv zu bewertende Dienstleistung. Varianten dieses Angriffs sind:
   (a) Das „Weiterleiten" erfundener bzw. Wiederholen eines Pakets führt zwar zu positiven Bewertungen, ergibt für den Angreifer jedoch keinen Energievorteil.
   (b) Unterhält ein Knoten mehrere Identitäten und tritt er unter einer Identität als Weiterleiter der unter einer anderen Identität erzeugten Pakete auf, kann er positiven Bewertungen für eigene Pakete erhalten (attraktiver Angriff!).
3. *Weiterleitung über nicht erreichbare Knoten:* Dieser Angriff kann dazu verwendet werden, selbst eine positive Bewertung zu erhalten (auf Kosten eines Sendevorgangs) oder anderen Knoten negative Bewertungen zu verursachen.
4. *Senden an einen falschen nächsten Weiterleiter:* Der Angreifer kann evtl. den Aufwand der Teilnahme an einem Routing-Protokoll einsparen und z. B. immer an einen zufällig gewählten Nachbarn weiterleiten, oder an einen bestimmten, besonders zu strapazierenden Nachbarn, um diesem zu schaden. Solche Weiterleitung ist keine vollwertige Leistung.
5. *Vortäuschen, ein weiterzuleitendes Paket nicht erhalten zu haben:* Dieser Angriff kann dazu verwendet werden, negative Bewertung für Nichtweiterleitung zu vermeiden, ist jedoch nur durchführbar, wenn die negative Bewertung noch davon abhängig ist, ob der Beobachter eine im verwendeten Schicht-2-Protokoll vorgesehene Empfangsbestätigung beobachtet, die der Angreifer dann absichtlich unterschlagen kann; in diesem Fall handelt es sich um einen attraktiven Angriff.
6. *Vorschieben einer negativen Zugangsentscheidung als Begründung für Nichtweiterleitung:* Damit Beobachter nicht von böswillig unterlassener Weiterleitung ausgehen, werden negative Zugangsentscheidungen bekannt gemacht. Unkooperative Knoten könnten ausschließlich negative Entscheidungen treffen.

Gegenmaßnahmen Gegen Angriffe, durch welche der Angreifer sich einen Vorteil verschaffen kann, werden die folgenden Maßnahmen vorgeschlagen. Rein destruktive Angriffe ohne umfassende Auswirkungen werden nicht betrachtet.

- *Werte nur dann positiv, wenn der vorige Weiterleitungsschritt beobachtet wurde:* Anhand der Beobachtung des vorigen Weiterleitungsschrittes kann erkannt werden, dass ein Paket weder erfunden noch wiederholt wurde. Jeder Knoten speichert

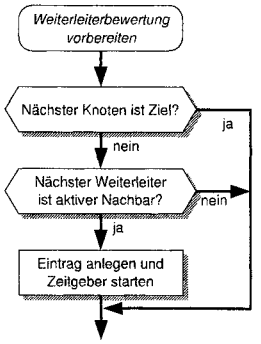


Abb. 2. Vorbereitung der Weiterleitungsbeobachtung

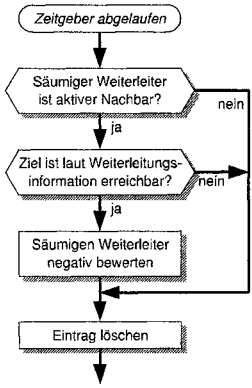


Abb. 3. Negative Bewertung bei Ablauf des Zeitgebers

dafür zu jedem beobachteten Paket für kurze Zeit einen Hash-Wert, anhand dessen er wiedererkennen kann, ob es vom nächsten Weiterleiter erneut ausgesendet wird. Für die Bewertung fallen Beobachter weg, die den vorigen Weiterleitungsschritt nicht beobachten konnten, so dass insgesamt weniger positive Bewertungen vergeben werden können (richtet sich gegen Angriff 2).

- *Werte nur dann negativ, wenn der Knoten, der weiterleiten soll, in der Nachbarschaft des Beobachters ist:* Diese Maßnahme richtet sich gegen die Angriffe 1 und 3 bzgl. negativer Bewertung und stellt weiterhin sicher, dass Knoten nur dann negativ bewertet werden, wenn eine potentielle Weiterleitung überhaupt vom bewertenden Knoten beobachtet werden könnte.
- *Werte bei ausbleibender Weiterleitung auch dann negativ, wenn der Empfang des weiterzuleitenden Pakets vom säumigen Weiterleiter nicht bestätigt wurde:* Diese Maßnahme hilft bei im Schicht-2-Protokoll vorgesehenen Empfangsbestätigungen gegen Vortäuschen von Empfangsproblemen (Angriff 5).
- *Übertrage in Fehlermeldungen aufgrund von Zugangsentscheidungen auch die abgelehnten Pakete:* Negative Zugangsentscheidungen sind somit mindestens so umfangreich wie die eigentlich weiterzuleitenden Pakete so dass Angriff 6 keinen Energievorteil mehr ergibt. Weiterhin kann dadurch jeder Beobachter nach Empfang der Fehlermeldung das abgelehnte Paket in seiner Liste löschen.

3.3 Detaillierte Verfahrensbeschreibung

Beim Absenden selbst erzeugter Pakete sowie bei der Weiterleitung fremder Pakete wird kurz vor der Aussendung zum nächsten Weiterleiter die Beobachtung vorbereitet (siehe Abbildung 2). Dazu wird das Paket in die Liste beobachteter Pakete aufgenommen, falls es erstens nach der selbst durchzuführenden Weiterleitung tatsächlich nochmal weitergeleitet werden muss (nächster Knoten $\neq$ Zielknoten) und sich zweitens der nächste Weiterleiter in der Nachbarschaft des Beobachters befindet. Der angelegte Eintrag enthält einen Hashwert über alle Bestandteile des Pakets, die bei der Weiterleitung

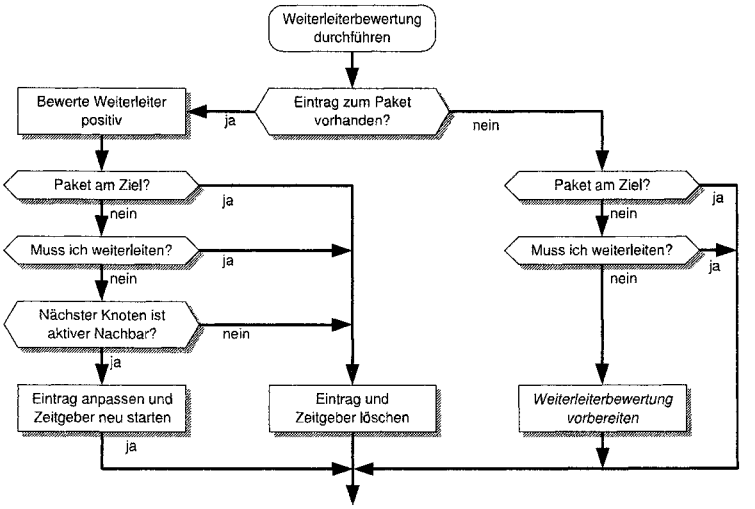


Abb. 4. Ablauf der Weiterleitungsbeobachtung

nicht verändert werden, sowie die Adresse des Schicht-2-Empfängers. Dem Eintrag ist ein Zeitgeber zugeordnet, bis wann die nächste Weiterleitung erfolgt sein sollte.

Abbildung 4 stellt die Abläufe dar, die für jedes empfangene oder mitgehörte Paket stattfinden. Anhand der Liste beobachteter Pakete wird festgestellt, ob das Paket bereits früher beobachtet bzw. selbst gesendet wurde. Hierzu wird der über das beobachtete Paket berechnete Hash-Wert in der Liste gesucht und geprüft, ob der Schicht-2-Absender des beobachteten Pakets mit dem im Eintrag gespeicherten Schicht-2-Empfänger übereinstimmt. In diesem Fall wird eine positive Beobachtung in dem Vertrauensprofil des Weiterleiters registriert. Unabhängig davon, ob eine positive Bewertung stattgefunden hat, ist das Ziel der nachfolgenden Schritte, die weitere Beobachtung des Pakets vorzubereiten, falls es nochmals weitergeleitet werden muss. Die Bearbeitung ähnelt deshalb der aus Abbildung 2, mit zwei Unterschieden: Muss der bearbeitende Knoten das Paket selbst weiterleiten, wird der Eintrag in der Liste beobachteter Pakete gelöscht, da das eigene Verhalten ja nicht bewertet wird. Weiterhin ist in manchen Fällen ggf. schon ein Eintrag vorhanden, der nur noch angepasst werden muss.

Schließlich muss noch der Fall eines ablaufenden Zeitgebers behandelt werden (Abbildung 3). Eine negative Bewertung wird in diesem Fall nur dann durchgeführt, wenn angenommen werden kann, dass der säumige Weiterleiter sich noch in der Nachbarschaft des Beobachters aufhält und das Ziel des Pakets erreichbar ist. Diese beiden Fragen werden anhand der lokalen Nachbarschaftsinformation und Routing-Tabelle geklärt. Sind die Bedingungen für eine negative Bewertung erfüllt, so wird aus der im Eintrag gespeicherten Schicht-2-Empfängeradresse die zugehörige Teilnehmerkennung ermittelt und für diese eine negative Beobachtung registriert. Der Eintrag in der Liste beobachteter Pakete wird abschließend gelöscht, da davon ausgegangen wird, dass das Paket entweder verloren gegangen ist oder den vom Beobachter wahrnehmbaren Bereich des Netzes verlassen hat.

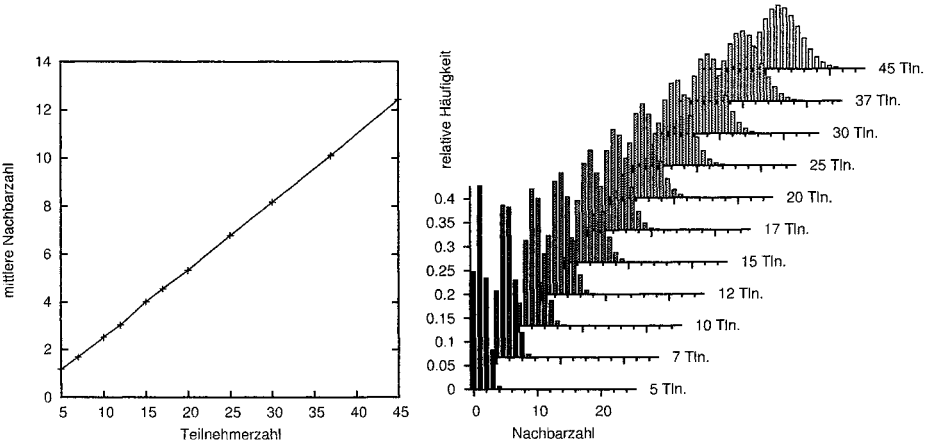


Abb. 5. Mittlere Nachbarzahl (links) und relative Häufigkeiten bestimmter Nachbarzahlen (rechts) in Abhängigkeit von der Teilnehmerzahl bei gleichbleibend großem Simulationsgebiet

4 Evaluation

Im Folgenden wird untersucht, ob und unter welchen Bedingungen die beschriebene Methode zur Beobachtung des Weiterleitungsverhaltens benachbarter Knoten ihre Funktion erfüllen und möglichst schnell eine ausreichende Menge an Information über das Verhalten der Nachbarn liefern kann. Hierfür wurde ein Netzknotenmodell innerhalb der Simulationsumgebung OMNeT++implementiert, wobei in der Schicht 2 zunächst ein idealisiertes Medienzugangsverfahren verwendet wurde, welches Kollisionen vermeidet und die verfügbare Bandbreite möglichst gut ausnutzt.

Zunächst wird die Verteilung von Nachbarzahl und Dauer von Nachbarschaftsverhältnissen in simulierten Szenarien betrachtet, da die Zahl der Nachbarn eines Knotens bestimmt, wie oft eine erbrachte Leistung oder ihre Verweigerung maximal beobachtet werden kann, und gegenseitige in Beobachtungen gewonnene Einschätzungen umso sicherer werden, je länger zwei Knoten Nachbarn sind.

Abbildung 5 zeigt links die per Simulation ermittelte Abhängigkeit zwischen Teilnehmerzahl und mittlerer Nachbarzahl bei gleichbleibender Gebietsgröße, rechts sind die relativen Häufigkeiten bestimmter Nachbarzahlen für Teilnehmerzahlen zwischen 5 und 45 aufgezeichnet. Dabei bewegten sich die Teilnehmer zufällig nach dem Random-Waypoint-Modell mit gleichverteilter Geschwindigkeit zwischen 1 und 10 m/s sowie gleichverteilter Wartezeit zwischen 1 und 30 s in einem torusförmigen Simulationsgebiet von 600 m Länge und 600 m Breite; als maximale Sendereichweite wurden 180 m angenommen. Es ist eine lineare Abhängigkeit zwischen Teilnehmerzahl und mittlerer Nachbarzahl zu erkennen. Theoretische Überlegungen zum Zusammenhang zwischen Gesamtteilnehmerzahl N_T, Gebietsgröße A, Übertragungsreichweite r und mittlerer Nachbarzahl N_N bestätigen dieses Ergebnis:

$$N_N = \frac{N_T - 1}{A} \cdot \pi \cdot r^2. \tag{1}$$

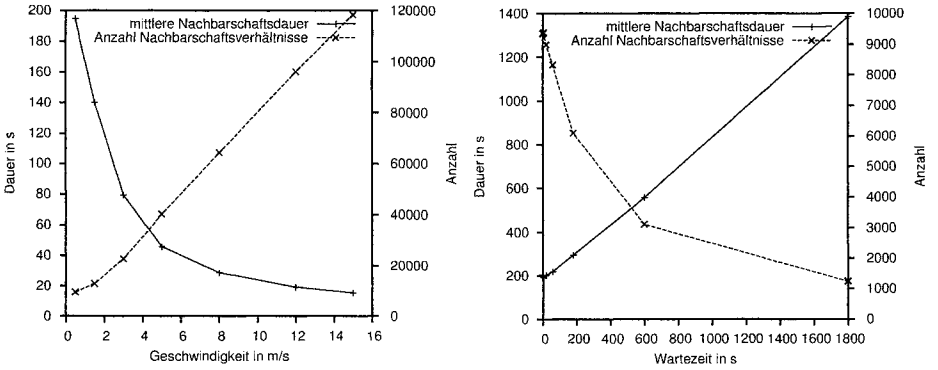


Abb. 6. Mittlere Dauer von Nachbarschaftsverhältnissen in Abhängigkeit von der Geschwindigkeit (Aufenthaltsdauer 1 s; links) bzw. Aufenthaltsdauer (Geschwindigkeit um 0,5 m/s; rechts)

Die maximale Übertragungsreichweite r, die in der Realität aufgrund unterschiedlicher Sendeleistung und Emfangsempfindlichkeit sowie von Störeinflüssen für jede Knotenpaarung verschieden sein kann, wird hierbei vereinfachend als eine scharfe, für alle Knotenpaarungen gleiche Entfernung angenommen. Fehlerfreie Signalübertragung sei genau bei jeder kürzeren Distanz zwischen zwei Knoten möglich.

Versuche mit unterschiedlicher Parametrisierung des Random-Waypoint-Modells zeigen, dass die mittlere Nachbarzahl stets dem rechnerisch ermittelten Wert entspricht.

Bei sehr kurzer Aufenthaltszeit (Mittelwert 1 s) ergibt sich ein regelmäßiger Zusammenhang zwischen Knotengeschwindigkeit und Dauer von Nachbarschaftsverhältnissen. Abbildung 6 stellt hierzu die mittlere Nachbarschaftsdauer und die dazu umgekehrt proportionale Anzahl von Nachbarschaftsverhältnissen über der gewählten mittleren Knotengeschwindigkeit dar. Bei längeren Aufenthaltszeiten streut die Dauer von Nachbarschaftsverhältnissen stärker und ist im Mittel größer. Variiert man statt der Geschwindigkeit die Aufenthaltsdauer, so erhält man den rechts dargestellten Verlauf der mittleren Nachbarschaftsdauer. Man sieht, dass der Mittelwert nahezu linear mit der Aufenthaltsdauer ansteigt. Zusammenfassend führen beim Random-Waypoint-Bewegungsmodell größere Geschwindigkeiten zu einer Häufung kurzer Nachbarschaftsdauern und größere Aufenthaltszeiten zu einer stärkeren Streuung.

Damit eine erbrachte Weiterleitungsleistung auch bewertet werden kann, müssen nicht nur Beobachter vorhanden sein, sondern es muss auch jeweils die Zusatzbedingung erfüllt sein, dass der Beobachter auch den vorigen Weiterleitungsschritt beobachtet hat, damit er sicher sein kann, dass es sich um eine Weiterleitung und nicht um die Aussendung einer eigenen Nachricht handelt (erste Gegenmaßname in Abschnitt 3.2). Um empirisch zu untersuchen, ob die Zusatzbedingung die Bewertungsmöglichkeiten zu stark einschränkt, wurde im Versuch zunächst die Anzahl aller vergebenen positiven Weiterleitungsbewertungen erfasst und in Verhältnis zur Zahl der insgesamt durchgeführten Weiterleitungsvorgänge gesetzt. Abbildung 7 zeigt links das Ergebnis, aufgetragen über der Teilnehmerzahl des verwendeten Netzes; zum Vergleich ist dort außerdem die ermittelte mittlere Nachbarzahl eingetragen. Man sieht, dass die mittlere

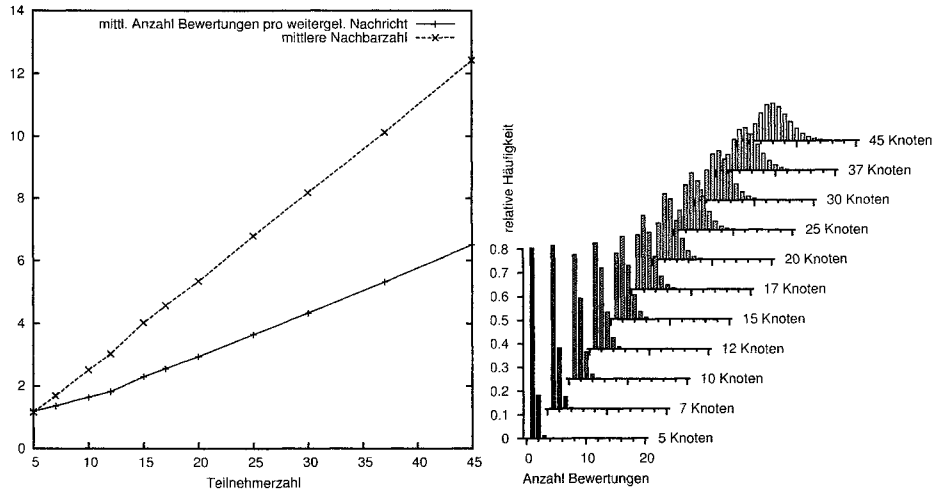


Abb. 7. Anzahl (positiver) Bewertungen pro weitergeleiteter Nachricht (links) sowie Häufigkeiten bestimmter Anzahlen positiver Bewertungen bei weitergeleiteten Nachrichten (rechts)

Zahl der Bewertungen pro weitergeleiteter Nachricht näherungsweise proportional zur Teilnehmer- und etwa halb so groß wie die mittlere Nachbarzahl ist, abgesehen vom unteren Teilnehmerzahlbereich, wo sie etwas höher ist. Bei der Anzahl von 25 Teilnehmern wurden immerhin 3,64 positive Bewertungen pro erfolgter Weiterleitung vergeben. Es erscheint somit durchaus plausibel, dass aufgrund der bewerteten Beobachtungen zügig Einschätzungen zur Kooperationsbereitschaft der beobachteten Teilnehmer gewonnen werden können. Bei diesen Experimenten wurden nur positive Bewertungen betrachtet, und die verwendeten Modellteilnehmer verweigerten niemals absichtlich Leistungen; andernfalls käme jeweils eine ähnliche Zahl an Negativbewertungen zustande. Eine Differenzierung wurde für eine erste Evaluierung nicht vorgenommen, da hierfür nur die Gesamtzahl verwertbarer Beobachtungen eingeschätzt werden muss.

Bezüglich des Einflusses der Teilnehmermobilität auf die in diesem Abschnitt beschriebenen Zusammenhänge ist festzustellen, dass sie lediglich von der Teilnehmerdichte des betrachteten Netzwerks abhängen. Da diese bei Verwendung des Random-Waypoint-Modell homogen ist, ergeben sich auch durch Variation der Verteilungen für Geschwindigkeiten und Aufenthaltsdauern keine Änderungen. Realitätsnähere Bewegungsmodelle bewirken stärkere Häufungen der Teilnehmer, was auch zu einer höheren mittleren Nachbarzahl führt. Versuche mit solchen Mobilitätsmodellen mit inhomogeneren Teilnehmerdichten ergeben regelmäßig eine höhere Bewertungsquote, da hier mehr Nachbarschaftsbeziehungen bestehen und somit mehr Beobachtungen bewertet werden können.

5 Zusammenfassung und Ausblick

Der vorliegende Beitrag stellt ein Verfahren zur Beobachtung und Bewertung des Weiterleitungsverhaltens von Ad-hoc-Netz-Teilnehmern vor, welches im Unterschied zu

existierenden Verfahren erstens nicht von einem bestimmten Routing-Protokoll abhängig ist und zweitens auch Beobachter erlaubt, die selbst nicht am beobachteten Weiterleitungsvorgang beteiligt sind. Gezielte Täuschungen des Verfahrens, durch welche Angreifer sich Vorteile durch falsche Bewertungen verschaffen könnten, wurden dabei nach einer detaillierten Analyse der Angriffsmöglichkeiten durch Gegenmaßnahmen ausgeschlossen.

Dadurch, dass Beobachtungen auch durch Teilnehmer erfolgen können, die an dem Weiterleitungsvorgang selbst nicht beteiligt sind, entstehen wesentlich mehr Beobachtungen pro Weiterleitungsvorgang als bei existierenden Verfahren, wo nur jeweils eine Beobachtung anfallen kann. Die simulative Evaluation des Verfahrens zeigt, dass die Anzahl verwertbarer Beobachtungen etwa halb so groß ist wie die Zahl der Nachbarn des weiterleitenden Knotens. Diese höhere Beobachtungszahl führt insgesamt dazu, dass in kürzerer Zeit bessere Einschätzungen ermittelt werden können.

Einschätzungen beliebiger entfernter Netzknoten, wie sie etwa für eine Zugangskontrolle aufgrund der Einschätzung der Quellknoten weiterzuleitender Pakete benötigt werden, kann die Beobachtung von Nachbarn nur allmählich durch mobilitätsbedingte Durchmischung der Teilnehmer liefern. Wesentlich schneller erhält man Einschätzungen entfernter Teilnehmer, wenn lokal ermittelte Einschätzungen im Netz verteilt und von anderen Teilnehmern in geeigneter Weise einbezogen werden. Ein entsprechendes, auf dem hier vorgestellten Ansatz aufbauendes Verfahren wird in [Kraf06] beschrieben.

In den hier vorgestellten Simulationen wurde ein idealisiertes Medienzugangsverfahren verwendet. Bei realen Verfahren können durch gleichzeitiges Senden von Knoten, die sich außerhalb ihrer gegenseitigen Sendereichweite befinden, bei Beobachtern Überlagerungen entstehen, durch welche Beobachtungen verhindert werden. Die Untersuchung der Auswirkungen solcher Effekte ist Gegenstand zukünftiger Arbeiten.

Literaturverzeichnis

[BaBa03] S. Bansal und M. Baker. Observation-based cooperation enforcement in ad hoc networks. Technischer Bericht, Stanford University, 2003.

[BuBo02] S. Buchegger und J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks). In *Proc. ACM Symp. on Mobile Ad Hoc Networking & Computing (MobiHoc)*, Juni 2002.

[BuBo04] S. Buchegger und J.-Y. Le Boudec. A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In *Proc. 2nd Workshop on the Economics of Peer-to-Peer Systems*, Juni 2004.

[BuTLB04] S. Buchegger, C. Tissieres und J. Y. Le Boudec. A Test-Bed for Misbehavior Detection in Mobile Ad-hoc Networks - How Much Can Watchdogs Really Do? In *Proc. IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, English Lake District, UK, Dezember 2004.

[HeWK04] Q. He, D. Wu und P. Khosla. SORI: A secure and objective reputation-based incentive scheme for ad hoc networks. In *Proc. IEEE Wireless Communications and Networking Conference*, Atlanta, GA, USA, März 2004.

[Kraf06] D. Kraft. *Verteilte Zugangskontrolle in offenen Ad-hoc-Netzen.* Dissertation, Universität Karlsruhe (TH), 2006.

[MGLM00] S. Marti, T. J. Giuli, K. Lai und M.Baker. Mitigating Routing Misbehaviour in Mobile Ad hoc Networks. In *Proc. 6th International Conf. on Mob. Comp. and Networking (MOBICOM)*, August 2000, S. 255–265.

Teil IV

Dienstgüte und Sicherheit

A Priori Detection of Link Overload due to Network Failures

Jens Milbrandt, Michael Menth, and Frank Lehrieder

Department of Distributed Systems, Institute of Computer Science
University of Würzburg, Am Hubland, D-97074 Würzburg, Germany
Phone: (+49) 931-888 6644, Fax: (+49) 931-888 6632
{milbrandt,menth,lehrieder}@informatik.uni-wuerzburg.de

Abstract. Restoration or protection switching mechanisms are triggered by link or node failures to redirect traffic over backup paths. These paths then carry the normal and the backup traffic which may lead to overload and thereby to quality of service (QoS) violations, i.e. to excessive packet loss and delay. In this paper, we present a method to assess the potential overload of the links due to network failures. We calculate the complementary cumulative distribution function (CCDF) of the relative load for each link in the network. We discuss various performance measures that condense this information to a single value per link which is suitable for a link ranking. This helps to identify weak spots of the network and to appropriately upgrade the bandwidth of links although they are not overloaded during normal operation. We implemented the concept in a software tool which helps network providers to anticipate the potential overload in their networks prior to failures and intended modifications (new infrastructure, new routing, new customers, ...) and to take appropriate actions.

1 Introduction

Network resilience is an important issue in carrier grade networks. It comprises the maintenance of both the connectivity and the quality of service (QoS) in terms of packet loss and delay during network failures. To maintain connectivity, restoration or protection switching mechanisms are triggered by link or node failures and redirect traffic over backup paths. These paths then carry the normal and the backup traffic which may lead to overload and thereby to QoS violations. While the service availability in terms of connectivity has been studied quite well [1], the a priori detection of overload due to network failures has not yet been investigated in depth.

In this paper, we present a concept to calculate the risk of overload due to network failures and implement it in a software tool. Basically, failures are associated with certain probabilities and the triggered traffic redirection causes a specific relative link load on all links. Considering all possible network failures, the law of total probability allows the derivation of the complementary cumulative distribution function (CCDF) of

This work was funded by the Bavarian Ministry of Economic Affairs and the German Research Foundation (DFG). The authors alone are responsible for the content of the paper.

the relative load of all links. However, this simple principle comes with several problems that are solved in this paper.

Firstly, the evaluation of all possible failure scenarios is computationally not feasible for medium size or large networks. Therefore, the analysis must limit its scope to the most relevant failure scenarios. In most previous studies, only the impact of single failures has been considered as the probability of multiple failures is rather low. However, the number of multiple failures increases strongly with the network size such that their overall probability cannot be neglected for the approximation of the CCDF. We define all (multi-)failures with a probability larger than p_{min} as relevant – irrespective of their number of failed components – and consider them for our analysis. We present an algorithm to efficiently find the set of all relevant failure scenarios. The framework is designed in such a way that both independent and correlated multiple failures can be respected. The latter ones are better known as shared risk resource groups (SRRGs) [2]. For instance, shared risk link groups (SRLGs) consist of IP links that are logically distinct on the network layer but share a common resource on the link layer; the failures of this common resource makes all links of the SRLG inoperative.

Secondly, the CCDF of the relative link load is the most detailed information we obtain from the analysis, but it is not suitable to compare the risk of link overload of several links because the CCDF does not define a strict order relation among them. To facilitate link rankings, we discuss various performance measures that condense the information of the CCDF into a single value. These condensed values help operators to quickly identify the most critical links and to find weak spots in their networks. It allows them to prevent congestion due to redirected traffic by upgrading the bandwidth of the most jeopardized links appropriately although they are not overloaded during normal operation.

This paper is structured as follows. In Section 2 we review related work regarding network resilience. Section 3 explains our algorithms to calculate the CCDFs of the relative link loads that are caused by the most relevant failures. Section 4 visualizes these CCDFs and develops simple assessment functions to condense their information into a single value; furthermore, additional features of our software tool are illustrated to provide a quick overview of the potential overload in the entire network. Finally, Section 5 summarizes this work and gives an outlook on possible extensions.

2 Network Failures and Resilience

In this section, we review basics about network failures and resilience mechanisms that deviate the traffic around an outage location in the network. We give an overview on similar work and comment on our contribution.

2.1 Network Failures

A good overview and characterization of network failures is given in [3, 4]. We can distinguish planned and unplanned failures. Planned outages are intentional, e.g. due to maintenance, and operators can take countermeasures in advance. Unplanned outages are hard to predict and can be further subdivided into failures with internal causes (e.g. software bugs, component defects, etc.) and those with external causes (e.g. digging works, natural disasters, etc.).

Quantitative analyses and statistics about frequency and duration of failure events that occur in operational networks like the Sprint IP backbone are given in [5, 6]. They show that link failures are part of everyday's network operation and the majority of them is short-lived, i.e., their duration is shorter than 10 minutes. Moreover, they indicate that 20% of all failures are due to planned maintenance activities. Of the unplanned failures, almost 30% are shared by multiple links and are related to problems with routers or optical equipment, while 70% affect only a single link at a time.

The mean time between failures (MTBF) and the mean time to repair (MTTR) are used to characterize the unavailability of a network element by $p = \frac{\text{MTTR}}{\text{MTBF}}$. Different values for MTBF and MTTR can be found in the literature for nodes and for links [3, 4, 7–9]. In this study, we choose MTTR = 2 h and MTBF = $2 \cdot 10^6$ h for nodes, i.e. $p_{node} = 10^{-6}$. Furthermore, we use MTTR = 12 h and MTBF = $\frac{300\,\text{km}}{L(l)} \cdot 365 \cdot 24$ h for links where $L(l)$ denotes the length of the link l such that we get $p_{link} = L(l)/219000$ km.

2.2 Resilience Mechanisms

In case of a network failure, resilience mechanisms redirect the affected traffic around the failure location. They can be classified into protection switching and restoration. Protection switching establishes backup paths in advance while restoration finds a new path after a failure has occurred. Therefore, protection switching reacts faster than restoration. A good overview of resilience mechanisms can be found in [3, 4]. In this study, we use IP rerouting for illustration purposes, but our framework does not depend on any specific routing or resilience mechanism.

IP networks implement destination based routing and calculate the routing tables in a distributed manner according to the shortest path principle. If a link or node fails, the routing tables are automatically recalculated and the traffic follows the next shortest paths after some time [10]. Thus, e2e IP connectivity is maintained as long as the network is physically connected. If several shortest paths exist towards a destination, the traffic may be forwarded to the interface with the highest priority, which is single shortest path (SSP) routing, or it may be split equally among all interfaces of the shortest paths, which is called equal-cost multipath (ECMP) routing. In our study, we use ECMP with the standard hop count metric, i.e., all link costs are set to one. However, the link costs may be manipulated for traffic engineering purposes, e.g., to minimize the link utilization under normal conditions [11] or to make the network robust against link failures [12–15].

2.3 Related Work Regarding Resilience Analysis

The authors of [16] present calculations for e2e availability of various resilience mechanisms, e.g. dedicated and shared primary and backup path concepts or restoration methods. When rerouting in networks is considered, many multiple failures affect the availability which leads to complex calculations. Therefore, either a limited number of most probable failure scenarios is taken into account [17] or the analysis is limited to only single or double failures. In [18–21] the impact of double failures is analyzed in networks that are resilient to single failures. Most papers regarding resilience issues consider only e2e availability [1], but some other studies also take the expected lost traffic (ELT) as a performance measure into account to quantify the missing capacity

during failures [7, 9]. To reduce ELT, backup capacity is required that may be used by low priority traffic during failure-free operation of the network [22]. Resilience can also be considered on the application layer, e.g., the availability of services can be improved by alternative servers and caching techniques [23]. NetScope is a tool to calculate the load on the links of a network to predict the effect of various traffic matrices, special failure scenarios, or alternate routing, and can be used for the inner loop of routing optimization [24]. Our tool is basically an extension of that approach towards statistical results.

2.4 Contribution of this Work

The above mentioned studies are static in the sense that they respect only explicitly specified failures of (single) network elements. This is a reasonable start for resilient QoS provisioning, but the probability of multiple network failures grows with increasing network size. Therefore, multiple failures need to be taken into account if the network size increases. The objective is to make networks resilient to the majority of likely failure scenarios in the sense that no overload occurs due to redirected traffic. Hence, the impact of the majority of likely failure scenarios is more important for the network resilience than the impact of a few devastating but very unlikely multiple failures.

The novelty of this work is the assessment of the potential overload in a network. We present a framework that yields a distribution of the link load caused by redirected traffic in failures scenarios. This helps Internet service providers (ISPs) (1) to detect weak spots in their network and (2) to improve the resilience of their network systematically without general overprovisioning [25]. The improvement can be achieved (a) by improved routing and rerouting in failure cases, (b) by the upgrade of existing links, or (c) by the introduction of new infrastructure. We currently develop a tool that predicts the resilience of the network after such modifications to support the ISP with his decision process.

3 Calculation of the Relative Link Load

We assess the potential overload of the links by analyzing the impact of failure scenarios on their relative link load. As not all failure scenarios can be covered by the analysis due to computational complexity, we determine the most relevant ones and take only them into account. Various failure scenarios lead to the same so-called "effective" working topology. We take advantage of that fact for the calculation of the traffic rates on the links. They are needed to derive the link-specific conditional complementary cumulative distribution function (CCDF) of the relative link load.

3.1 Relevant Failure Scenarios

In our study, we consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of routers $v \in \mathcal{V}$ and directed links $l \in \mathcal{E}$. A simple application of our framework assumes link and node failures as basic and independent failure events. However, it is possible to model failure events on an even lower level, e.g., individual interfaces and line cards. The set $\hat{\mathcal{S}}$ represents the set of all independent failure events. Note that this set may also contain shared risk resource groups (SRRGs) such as shared risk link or node groups (SRLG, SRNG) [2].

Each of these failure events occurs with probability $p(\hat{s})$ and we number the events $\hat{s}_i$ $(0 \leq i < |\hat{\mathcal{S}}|)$[1] in a descending order according to $p(\hat{s}_i)$. A (compound) failure scenario $s \subseteq \hat{\mathcal{S}}$ consists of several independent failure events $\hat{s} \in \hat{\mathcal{S}}$ that incidentally occur at the same time. Its probability is $p(s) = (\Pi_{\hat{s} \in s} p(\hat{s})) \cdot (\Pi_{\hat{s} \in \hat{\mathcal{S}} \setminus s}(1 - p(\hat{s})))$ and it is relevant if it has a probability of at least $p(s) \geq p_{min}$. Finally, the set of relevant failure scenarios $\mathcal{S} = \{s \in \mathcal{P}^{\hat{\mathcal{S}}} : p(s) \geq p_{min}\}$ comprises all relevant failure scenarios and is a subset of the power set $\mathcal{P}^{\hat{\mathcal{S}}}$. In particular, the failure-free scenario $s = \emptyset$ is relevant and part of $\mathcal{S}$.

Algorithm 1 constructs $\mathcal{S}$. At the beginning, the global set of relevant failure scenarios is initialized with $\mathcal{S} = \emptyset$. The recursive procedure RELEVANTSCENARIOS$(i, s, p(s))$ is called with arguments $(0, \emptyset, 1)$. The algorithm steps recursively through the set of independent failure events $\hat{s}_i \in \hat{\mathcal{S}}$. It constructs a compound failure scenario s incrementally and the recursion ends either if the probability $p(s)$ of the partial compound failure scenario s is lower than p_{min} or if all independent failure events $\hat{s}_i \in \hat{\mathcal{S}}$ have been considered as potential members of s. In the latter case, the failure scenario s joins $\mathcal{S}$ at the end of each recursion. At program termination, the set $\mathcal{S}$ contains all compound failure scenarios with a probability of at least p_{min}.

Input: failure event number i, partial scenario s, and its probability $p(s)$
  if $(i = |\hat{\mathcal{S}}|)$ **then** {all independent failure scenarios have been considered}
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$
  else {partial scenario s is probable enough to be relevant}
    if $(p(s) \cdot p(\hat{s}_i) > p_{min})$ then RELEVANTSCENARIOS$(i+1, s \cup \hat{s}_i, p(s) \cdot p(\hat{s}_i))$
    if $(p(s) \cdot (1 - p(\hat{s}_i)) > p_{min})$ then RELEVANTSCENARIOS$(i+1, s, p(s) \cdot (1 - p(\hat{s}_i)))$
  end if

Algorithm 1: RELEVANTSCENARIOS: constructs the set of relevant scenarios $\mathcal{S}$.

3.2 Effective Topologies

The effective topology $T(s)$ caused by a compound failure scenario s is characterized by its set of working links and nodes. A link works if and only if itself and its adjacent routers do not fail. A router works if and only if itself and at least one of its adjacent links do not fail. Thus, all scenarios containing the failure of a router and some of its adjacent links lead to the same effective topology T. We subsume all of these scenarios in the set $\mathcal{S}(T)$ and the probability of T is inherited by $p(T) = \sum_{s \in \mathcal{S}(T)} p(s)$. The set $\mathcal{T} = \bigcup_{s \in \mathcal{S}} T(s)$ denotes the set of all relevant effective topologies.

3.3 Calculation of the CCDF of the Relative Link Load

Network failures lead to rerouting and changed load situations on many links. To detect the risk of potential overload due to failures, we derive the CCDF of the relative loads on all links based on the relevant failure scenarios $s \in \mathcal{S}$ and their probabilities $p(s)$. An aggregate g symbolizes the traffic between two specific routers and $\mathcal{G}$ is the set of all aggregates. The rate of a single aggregate is $c(g)$ and it is given by the traffic matrix. The routing function $u(T, l, g)$ determines the fraction of the rate $c(g)$ of aggregate g

[1] $|\mathcal{X}|$ denotes the cardinality of a set $\mathcal{X}$.

that flows over link l in the effective topology T. It allows the computation of the traffic rate on link l in the presence of effective topology T by $c(T,l) = \sum_{g \in \mathcal{G}} c(g) \cdot u(T,l,g)$.

Input: set of effective topologies $\mathcal{T}$
  for all $T \in \mathcal{T}$ **do**
    CALCULATEROUTING(T)
    for all $l \in \mathcal{E}$ **do**
      $c(T,l) \leftarrow 0$ {initialization}
      for all $g \in \mathcal{G}$ **do**
        $c(T,l) \leftarrow c(T,l) + c(g) \cdot u(T,l,g)$
      end for
      $\mathcal{L}(l) \leftarrow \mathcal{L}(l) \cup (T, c(T,l))$
    end for
  end for
Output: link-specific load sets $\mathcal{L}(l)$ for $l \in \mathcal{E}$

Algorithm 2: CALCULATELOAD: calculates the load $c(T,l)$ for each link $l \in \mathcal{E}$ and for all considered effective topologies $T \in \mathcal{T}$.

The load set $\mathcal{L}(l)$ contains all tuples $(T, c(T,l))$ consisting of the effective topologies $T \in \mathcal{T}$ and the corresponding traffic rates $c(T,l)$ on link l. Algorithm 2 computes the load sets $\mathcal{L}(l)$ for all links $l \in \mathcal{E}$ in an efficient way. The number of tuples $(T, c(T,l)) \in \mathcal{L}(l)$ depends on the set of relevant failure scenarios $\mathcal{S}$ which further depends on the threshold p_{min}. In particular, the probability of all relevant failure scenarios $p(\mathcal{S}) = \sum_{s \in \mathcal{S}} p(s)$ is smaller than 1 in most practical scenarios. Based on the traffic rates $c(T,l)$ and the capacity $c(l)$ of link l, its relative load $U(T,l) = \frac{c(T,l)}{c(l)}$ can be calculated. This link load $U(T,l)$ differs from a link utilization by the fact that it can take values larger than 1, neglecting that traffic may be lost due to congestion. It is useful to estimate the link bandwidth required to carry the traffic without any loss.

Finally, we can calculate the *conditional* CCDF of the relative link load by

$$p(U(l) > u | \mathcal{S}) = \frac{1}{p(\mathcal{S})} \cdot \sum_{\{s: s \in \mathcal{S} \wedge U(T(s),l) > u\}} p(s). \tag{1}$$

We can also give a lower and upper bound for the *unconditioned* CCDF of $U(l)$ by $p^{lower}_{bound}(U(l) > u) = p(U(l) > u | \mathcal{S}) \cdot p(\mathcal{S})$ and $p^{upper}_{bound}(U(l) > u) = p(U(l) > u | \mathcal{S}) \cdot p(\mathcal{S}) + (1 - p(\mathcal{S}))$.

4 Application Examples

We illustrate the above presented concept in an example network. We show the impact of the probability threshold p_{min} on the trustworthiness of the obtained CCDFs of the relative link load and demonstrate that the CCDFs do not establish an absolute order among the links. Therefore, we introduce different assessment functions that condense the information of the CCDF into a single value to get a simple result for the risk of

overload and to facilitate a comparison among links. The assessment functions also help to visualize the potential overload of the entire network at a glance. Finally, we give some examples for the applicability of the analysis in practice.

4.1 Test Environment

In the following, we apply the above presented analysis to the topology depicted in Figure 3 which is the basic structure of a typical core network in the U.S. There is one traffic aggregate $g = (v, w)$ for each pair of nodes v and w, and we define a static aggregate rate

$$c(g) = c(v, w) = \begin{cases} \frac{\pi(v) \cdot \pi(w) \cdot C}{Sum_{x,y \in \mathcal{V}, x \neq y} \pi(x) \cdot \pi(y)} & \text{if } v \neq w \\ 0 & \text{if } v = w \end{cases} \quad (2)$$

where $\pi(v)$ is the population of city $v \in \mathcal{V}$ and C is the rate of the overall network traffic. The populations for all cities associated with the nodes in our test network are taken from [26].

We assume hop-count based shortest path routing and rerouting using the equal-cost multipath (ECMP) option. We dimension the link capacities of our test network such that they are utilized by 20% in the non-failure case. Therefore, the choice of the overall rate C is irrelevant as we look only at the relative link load. This is an artificial scenario as it disregards available granularities for link capacities. However, we use this artificial setting only to illustrate our framework and we do not derive any results that are biased by this simplifying assumption.

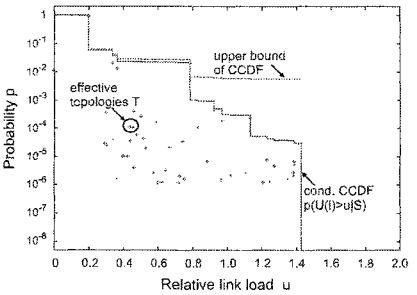
For the sake of simplicity, we limit our view to bidirectional link failures and node failures as basic failure events $\hat{s}$. We use the unavailability values given in Section 2.1 as failure probabilities $p(\hat{s})$. However, our tool is able to handle also more detailed failures such as those of line cards or single interfaces. In addition, SRRGs can also be modelled.

4.2 CCDF of the Relative Link Load

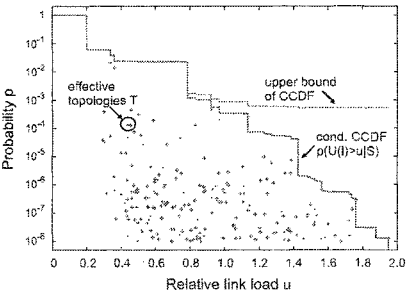
We first study the impact of the probability threshold p_{min} that controls the set of relevant failure scenarios $\mathcal{S}$ taken into account in the analysis. Then, we compare the CCDF for different links and show that it is not possible to establish a link order based on the CCDF of the relative link load.

Impact of the Probability Threshold p_{min} on the CCDF Figure 1(a) shows the conditional CCDF of the relative link load for link Dal→Was. The x-axis indicates the relative link load u and the logarithmic y-axis the probability $p(U(l) > u|\mathcal{S})$ that this value is exceeded. The points below the curve represent the effective topologies $T \in \mathcal{T}(\mathcal{S})$ that result from the relevant scenarios $\mathcal{S}$ and that cause the decay of the CCDF. Their coordinates consist of the relative link loads $U(T, l)$ and the probability $p(T)$. In our software the points for the effective topologies are sensitive such that the set of subsumed failure scenarios is displayed when the mouse is dragged over them.

The curve in Figure 1(a) is calculated based on a threshold of $p_{min} = 10^{-6}$ which leads to a set of $|\mathcal{S}| = 3260$ relevant failure scenarios with an overall probability of $p(\mathcal{S}) = 0.994504$. The graph also shows the lower and upper bound for the unconditioned CCDF. The probability threshold $p_{min} = 10^{-6}$ leaves a large uncertainty regarding



(a) $p_{min} = 10^{-6} \rightarrow$
$|\mathcal{S}| = 3260, \quad p(\mathcal{S}) = 0.994504.$



(b) $p_{min} = 10^{-8} \rightarrow$
$|\mathcal{S}| = 42030, \quad p(\mathcal{S}) = 0.999475.$

Fig. 1. Conditional CCDF of the relative link load $U(l)$ for link Dal→Was together with the lower and upper bound for the unconditioned CCDF.

the unconditioned CCDF in the range of interest where the link tends to be overloaded. Therefore, we plot the CCDF for $p_{min} = 10^{-8}$ in Figure 1(b). As a consequence, the set of relevant failure scenarios is now significantly larger such that it covers a probability of $p(\mathcal{S}) = 0.999475$. The curve has now a different shape in the right part of the graph and the distance between upper and lower bound for the conditioned CCDF is significantly smaller. In the following we use $p_{min} = 10^{-8}$.

Comparison of the CCDFs for Different Links The conditional CCDF of the relative link load $U(l)$ of a link l contains the maximum information about its overload probability. If the CCDF $p(U(l_0) > u|\mathcal{S})$ of the relative load of a link l_0 lies for all values below the one of another link l_1, then the risk of overload for l_0 is clearly smaller than for l_1. However, Figure 2 shows that link Dal→Was has a lower CCDF value than link SaF→Was for some relative link load values u (e.g. $u = 0.4$), and for some other values this is vice-versa (e.g. $u = 0.6$). Therefore, the CCDF does not provide an order relation for links and it is not appropriate for rankings of the links according to their potential overload.

4.3 Simple Assessment Functions for Potential Overload

The objective of our resilience analysis is to identify links that are most likely to be overloaded, but the CCDF cannot achieve that goal. Therefore, we propose three different assessment functions $R(l)$ that condense the information of the CCDF to a single value. They may be used for link rankings and to identify the most critical links.

Assessment Function Based on Overload Probabilities The network provider can define a critical relative link load value u_c that should not be exceeded. Thus, we define the assessment function for potential overload on link l by $R_{u_c}(l) = p(U(l) > u_c|\mathcal{S})$. Note that this ranking depends on the critical relative link load value u_c. Table 1 presents the rankings for $u_c \in \{0.3, 0.6, 0.9\}$ and shows that the value u_c indeed influences the ranking for the selected links.

Assessment Function Based on Relative Link Load Percentiles Another assessment function uses percentiles of the relative link load, i.e. the relative link load values

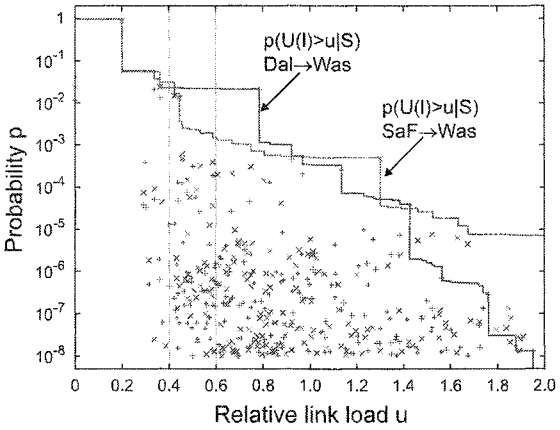


Fig. 2. Conditional CCDF of the relative link load for link Dal→Was and SaF→Was for $p_{min} = 10^{-8}$.

Table 1. Links ranked according to the overload probability $R_{u_c}(l)$.

link id	$R_{u_c}(l)$, $u_c = 0.3$	link id	$R_{u_c}(l)$, $u_c = 0.6$	link id	$R_{u_c}(l)$, $u_c = 0.9$
SaF-Sea	0.0734	LoA-SaF	0.0329	LoA-SaF	0.0324
LoA-SaF	0.0600	Den-SaF	0.0318	SaF-Sea	0.0285
Den-SaF	0.0540	SaF-Sea	0.0291	Den-SaF	0.0252

$R_q(l) = min(u : p(U(l) \leq u|\mathcal{S}) \geq q)$. It depends on the percentile parameter $0 \leq q \leq 1$. Table 2 shows the rankings for $q \in \{0.999, 0.99999\}$ and makes the dependency of R_{u_c} on the percentile parameter q obvious for selected links.

Table 2. Links ranked according to the relative link load percentile $R_q(l)$.

link id	$R_q(l)$, $q = 0.999$	link id	$R_q(l)$, $q = 0.99999$
Sea-Tor	1.512	NeO-Orl	8.761
Kan-SaF	1.3	Kan-SaF	2.628
NeO-Orl	0.2	Sea-Tor	2.198

Assessment Function Based on Weighted Relative Link Loads The above overload measures consider only a single point within the conditional CCDF of the relative link load $U(l)$, but operators might wish to take the information of the entire CCDF into account. We achieve this by weighting the CCDF with a suitable weight function $w(u)$:

$$R_w(l) = \int_0^{u_{max}} p(U(l) > u|\mathcal{S}) \cdot w(u) du \tag{3}$$

and we choose $w(u) = 10^{e_{mlwd} \frac{u}{u_{max}}}$ whereby e_{mlwd} is the maximum logarithmic weight difference. This assessment function respects all relative link load values up to u_{max} in the diagram. Thus, the ranking depends on u_{max} and e_{mlwd}. Table 3 shows the rankings

for $u_{max} = 1$ and $e_{mlwd} \in \{2, 4\}$ and makes the influence of the latter parameter explicit for selected links.

Table 3. Links ranked according to their weighted relative link load $R_w(l)$.

link id	$R_w(l)$, $e_{mlwd} = 2$	link id	$R_w(l)$, $e_{mlwd} = 4$
Dal-Was	0.514	Mia-Orl	6.7
Sea-Tor	0.511	Sea-Tor	5.926
Mia-Orl	0.478	Dal-Was	4.526

4.4 Potential Overload at a Glance

We show the potential overload in the network at a glance by displaying its topology with gray shaded links. The different intensities indicate the risk of overload due to network failures and, therefore, the value of the assessment function $R(l)$ is translated into an intensity value. A slide bar for the parameter u_c, q, or e_{mlwd} allows to change the intensities to get the suitable contrast. This is well feasible in realtime since the calculation of the above assessment functions needs only the stored CCDF but no further time-consuming analysis. Figure 3 illustrates the concept for the assessment function based on overload probabilities with $u_c = 0.6$.

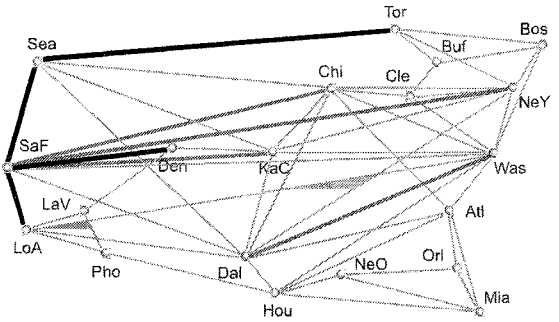


Fig. 3. The contrast of the links indicates their risk of overload due to network failures: dark links are more likely to be overloaded than light links.

4.5 Application Scenarios for A Priori Detection of Link Overload

Our tool allows a network operator to detect link overload a priori, i.e. before congestion occurs in the network due to redirected traffic in failure cases, and helps them to dimension the link bandwidths large enough to support QoS also during local network outages. The tool may be applied before the network configuration is changed to anticipate the impact of the changes on the potential overload. We list some examples of such changes.

- The tool helps to analyze whether a planned bandwidth upgrade of a link is sufficient to improve its potential overload or whether the upgrade is even wasteful.
- When new links or nodes are added to the network, the routing changes in the failure-free scenario and also in failure scenarios which impacts the potential overload on the links.

- When link metrics are changed, the routing changes both in the failure-free scenario and in failure scenarios which also impacts the potential overload on the links. It also helps to optimize link metrics manually.
- When new customers are added, the traffic matrix changes which impacts the relative load of the links and thereby their potential overload in the network.
- SRLGs of leased lines are not always known in advance. If new knowledge about them is available, a new failure event $\hat{s}$ is added that entails the simultaneous failure of the links in the SRLG. This may have a tremendous effect on the potential overload.

5 Conclusion

Network failures trigger restoration or protection switching mechanisms that redirect traffic onto backup paths which increases the relative load of their links. In this paper, we have proposed a framework to assess the risk of link overload in a network due to failures that occur with certain probabilities. We implemented a tool that derives an approximative complementary cumulative distribution function (CCDF) of the relative link load for all links in the network. The analysis considers only the set of relevant failure scenarios which have a probability larger than a minimum threshold p_{min}.

As the full information of the CCDF is often too complex for practical applications, we proposed to condense it to a single value by so-called risk assessment functions for which we discussed three basically different approaches. Their outcome can be used to rank links according to their risk of overload. These values lead to a presentation of the analysis results that is useful for network operators to quickly identify the links that have the highest risk to be overloaded although they do not reveal problems in the failure-free case. These links may be upgraded with additional bandwidth to prevent congestion due to redirected traffic in advance.

Currently, we extend our tool to a priori detect overload which may also be due to other causes such as local hot spots [25] or inter-domain rerouting [27].

References

1. Milbrandt, J., Martin, R., Menth, M., Hoehn, F.: Risk Assessment of End-to-End Disconnection in IP Networks due to Network Failures. In: 6^{th}IEEE Workshop on IP Operations and Management (IPOM), Dublin, Ireland (2006)
2. Datta, P., Somani, A.K.: Diverse Routing for Shared Risk Resource Groups (SRRG's) in WDM Optical Networks. In: 1^{st}IEEE International Conference on Broadband Communication, Networks, and Systems (BROADNETS). (2004) 120 – 129
3. Vasseur, J.P., Pickavet, M., Demeester, P.: Network Recovery. 1. edn. Morgan Kaufmann / Elsevier (2004)
4. Mukherjee, B.: Optical WDM Networks. 2 edn. Springer (2006)
5. Iannaccone, G., Chuah, C.N., Mortier, R., Bhattacharyya, S., Diot, C.: Analysis of Link Failures in an IP Backbone. In: ACM SIGCOMM Internet Measurement Workshop, Marseille, France (2002) 237 – 242
6. Markopoulou, A., Iannaccone, G., Bhattacharyya, S., Chuah, C.N.: Characterization of Failures in an IP Backbone. In: IEEE Infocom, Hongkong (2004)
7. Willems, G., Arijs, P., Parys, W.V., Demeester, P.: Capacity vs. Availability Trade-offs in Mesh-Restorable WDM Networks. In: International Workshop on the Design of Reliable Communication Networks (DRCN), Budapest, Hungary (2001)

8. Cankaya, H.C., Lardies, A., Ester, G.W.: A Methodology for Availability-Aware Cost Modelling of Long-Haul Networks. In: International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), San Jose, CA (2004)
9. Maesschalck, S.D., Colle, D., Lievens, I., Pickavet, M., Demeester, P., Mauz, C., Jaeger, M., Inkret, R., Mikac, B., Derkacz, J.: Pan-European Optical Transport Networks: an Availability-Based Comparison. Photonic Network Communications **5** (2005) 203–225
10. Iannaccone, G., Chuah, C.N., Bhattacharyya, S., Diot, C.: Feasibility of IP Restoration in a Tier-1 Backbone. IEEE Network Magazine (Special Issue on Protection, Restoration and Disaster Recovery) (2004)
11. Fortz, B., Rexford, J., Thorup, M.: Traffic Engineering with Traditional IP Routing Protocols. IEEE Communications Magazine **40** (2002) 118–124
12. Fortz, B., Thorup, M.: Robust Optimization of OSPF/IS-IS Weights. In: International Network Optimization Conference (INOC), Paris, France (2003) 225–230
13. Nucci, A., Schroeder, B., Bhattacharyya, S., Taft, N., Diot, C.: IGP Link Weight Assignment for Transient Link Failures. In: 18^{th} International Teletraffic Congress (ITC), Berlin (2003)
14. Yuan, D.: A Bi-Criteria Optimization Approach for Robust OSPF Routing. In: 3^{rd}IEEE Workshop on IP Operations and Management (IPOM), Kansas City, MO (2003) 91 – 98
15. Sridharan, A., Guerin, R.: Making IGP Routing Robust to Link Failures. In: IFIP-TC6 Networking Conference (Networking), Ontario, Canada (2005)
16. Cholda, P., Jajszczyk, A.: Availability Assessment of Resilient Networks. In: 12^{th}GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB) together with 3^{rd}Polish-German Teletraffic Symposium (PGTS), Dresden, Germany (2004) 389–398
17. Li, V.O.K., Silvester, J.A.: Performance Analysis of Networks with Unreliable Components. IEEE Transactions on Communications **32** (1984) 1105–1110
18. Clouqueur, M., Grover, W.D.: Computational and Design Studies on the Unavailability of Mesh-restorable Networks. In: International Workshop on the Design of Reliable Communication Networks (DRCN), Munich, Germany (2000) 181 – 186
19. Clouqueur, M., Grover, W.D.: Availability Analysis of Span-Restorable Mesh Networks. IEEE Journal on Selected Areas in Communications **20** (2002) 810 – 821
20. Schupke, D.A., Prinz, R.G.: Capacity Efficiency and Restorability of Path Protection and Rerouting in WDM Networks Subject to Dual Failures. Photonic Network Communications **8** (2004)
21. Menth, M., Martin, R., Spoerlein, U.: Impact of Unprotected Multi-Failures in Resilient SPM Networks: a Capacity Dimensioning Approach. In: IEEE Globecom, San Francisco, California, USA (2006)
22. Durvy, M., Diot, C., Taft, N., Thiran, P.: Network Availability Based Service Differentiation. In: 11^{th}IEEE International Workshop on Quality of Service (IWQoS), Berkeley, CA, USA (2003) 305–324
23. Dahlin, M., Chandra, B.B.V., Gao, L., Nayate, A.: End-to-End WAN Service Availability. IEEE/ACM Transactions on Networking **11** (2003) 300–313
24. Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J.: NetScope: Traffic engineering for IP Networks. IEEE Network Magazine (2000) 11–19
25. Menth, M., Martin, R., Charzinski, J.: Capacity Overprovisioning for Networks with Resilience Requirements. In: ACM SIGCOMM, Pisa, Italy (2006)
26. Menth, M.: Efficient Admission Control and Routing in Resilient Communication Networks. PhD thesis, University of Würzburg, Faculty of Computer Science, Am Hubland (2004)
27. Schwabe, T., Gruber, C.G.: Traffic Variations Caused by Inter-domain Re-routing. In: International Workshop on the Design of Reliable Communication Networks (DRCN), Ischia Island, Italy (2005)

Analysis of a Real-Time Network Using Statistical Network Calculus with Approximate Invariance of Effective Bandwidth

Kishore Angrishi, Shu Zhang, and Ulrich Killat

Institute for Communication Networks, 4-06,
Hamburg University of Technology,
Hamburg, Germany
{kishore.angrishi,s.zhang,killat}@tu-harburg.de
http://www.tu-harburg.de/et6

Abstract. The modern industrial network traffic consists of an increasing amount of soft real-time flows, which can tolerate very small delay and losses. Allowing probabilistic Quality of Service (QoS) violation for these flows can greatly help to improve resource utilization. However, the improvement depends on the statistical properties of competing independent flows. The notion of effective bandwidth summarizes the statistical properties and QoS requirements of a flow. This paper suggests the usage of effective bandwidth to describe the arrival process of the flow along its path in a network within the framework of the statistical network calculus. Further, we improve the existing formal relationship between effective envelope and effective bandwidth to achieve a better bound for the arrival process. This new approach enables efficient utilization of statistical multiplexing of independent flows both at the ingress of the network and along the path of the flow.

Key words: Network calculus, effective bandwidth, Quality-of-service, statistical multiplexing.

1 Introduction

Many industrial communication systems require transmitting real time traffic, as most of the industrial processes are mission critical. This traffic can be broadly classified as hard real-time traffic allowing no deadline violation and soft real-time traffic with an upper bound of deadline violation probability ε. There is a lot of work including the elegant theory of (deterministic) network calculus [3, 4] for the analysis of service guarantees in network. Network calculus takes an envelope approach to describe arrivals and services in a network. Strength of the deterministic network calculus is that it can be used to determine the delay and backlog over multiple network nodes.

The worst case analysis for hard real-time traffic may claim much more resources than necessary for soft real-time traffic flows which can tolerate small deadline violations. This motivates the probabilistic extension of the network

calculus, commonly referred to as "statistical network calculus". A significant step towards statistical network calculus is presented in [5], where the concept of effective envelopes is derived. The main advantage of statistical network calculus over its deterministic counterpart is its capability to utilize statistical multiplexing within a framework to profit from multiplexing gain. However, once the effective envelope is fixed at the network ingress no further multiplexing gain is feasible. This paper addresses this issue.

The concept of effective envelopes is further developed in [5, 2] where its relation to the theory of effective bandwidth [11, 3] is shown. Effective bandwidth is a statistical descriptor of an arrival process sharing a resource with other flows. The effective bandwidth captures the effect of the considered flow's rate fluctuation and burstiness and represents the rate at which the considered flow needs to be served according to certain quality of service demands. One of the important properties of effective bandwidth is its *(approximate) invariance* [10, 14] along the network even for a small number of soft real-time flows.

In this paper, we explore the invariance condition of effective bandwidth along a path in the network. This invariance property and the relationship between effective bandwidth and effective envelope [2] is used to describe the arrival processes in the network by its effective bandwidth within the framework of statistical network calculus. The benefit of using effective bandwidth to describe the arrival processes inside the network is the efficient utilization of independence between the competing flows both at network ingress and along the path of the flows, whereas using effective envelope this is only possible at the network ingress. To enhance the benefit of our approach, we propose a tighter relationship between effective envelope and effective bandwidth using the accurate large deviation estimate of loss asymptotics studied in [7, 1].

In the remainder of this paper, we analyze only soft real-time flows that have stationary independent increments. The discrete time $t \in N = \{0, 1, 2, \dots\}$ is used. We apply the convention that capital letters indicate random variables and stochastic sequences. The analysis described in this paper assumes the network to be feedforward with respect to source-destination pairs.

The paper is organized as follows. In section 2, a brief background on deterministic and statistical network calculus is provided. The *(approximate) invariance* property of the effective bandwidths is briefly discussed in section 3. The usage of effective envelopes is improved in the inner network using our approach which is illustrated with an example and corresponding numerical results in section 4. Section 5 summarizes the conclusions.

2 Deterministic and Statistical Network Calculus

In network calculus [3, 4] flow's arrival, flow's departure and the service it receives at a node are described by real valued cumulative functions $A(0,t)$, $B(0,t)$, and $F(0,t)$ (dynamic F-server [3]) respectively in an interval $(0,t]$. We assume that there are no arrivals in the interval $(-\infty, 0]$. Clearly $A(0,t)$, $B(0,t)$ and $F(0,t)$ are non-negative and non-decreasing in t. The amount of data seen in an interval (s,

t] is denoted by $A(s,t) = A(0,t) - A(0,s)$ and $B(s,t) = B(0,t) - B(0,s)$. Similar assumptions are not generally made for $F(s,t)$. $A(s,t)$, $B(s,t)$ and $F(s,t)$ are nonnegative, increasing in t, decreasing in s, and $A(t,t) = B(t,t) = F(t,t) = 0$ for all t. The foundations of network calculus are min-plus convolution and min-plus deconvolution. The operations of min-plus convolution ($\otimes$) and deconvolution ($\oslash$) of the arrival process $A(s,t)$ and dynamic F-server $F(s,t)$ [3] are defined for any $t \geq s \geq 0$ as

$$(A \otimes F)(s,t) = \inf_{\tau \in [s,t]} [A(s,\tau) + F(\tau,t)] \leq B(s,t) \tag{1}$$

$$(A \oslash F)(s,t) = \sup_{\tau \in [0,s]} [A(\tau,t) - F(\tau,s)] \geq B(s,t) \tag{2}$$

Note that the defined operators $\otimes$ and $\oslash$ are not commutative. Eq.(1) is the formal definition of dynamic F-server [3] and eq.(2) defines the output bound.

The applicability of network calculus suffers from the worst-case modeling that is used. Since traffic is statistically multiplexed at network nodes, such scenarios are extremely rare. Therefore, a probabilistic view, which considers that traffic in a packet network employing statistical multiplexing increases the achievable utilization in the network by tolerating rare adversarial events. The statistical network calculus seeks to quantify the statistical multiplexing gain while maintaining the algebraic aspects of the deterministic calculus. Statistical network calculus has been a hot topic recently and several groups have produced new results especially, [15, 16]. However, it can be shown that these results are directly related to the fundamental results presented in [5] and [6].

We follow the framework for a statistical network calculus presented in [5] and [6]. For traffic arrivals, we use a probabilistic measure called effective envelope [5]. An effective envelope for an arrival process A is defined as a non-negative function $\mathcal{G}^\varepsilon$ such that for all $t \geq s \geq 0$

$$P\{A(s,t) \leq \mathcal{G}^\varepsilon(s,t)\} \geq 1 - \varepsilon \tag{3}$$

In other words, an effective envelope provides a stationary bound for an arrival process. Effective envelopes can be obtained for individual flows, as well as for multiplexed arrivals [2]. To characterize the available service to a flow or a collection of flows we use effective service curves [6] which can be seen as a probabilistic measure of the available service. Given an arrival process A, an effective service curve is a non-negative function $\mathcal{F}^\varepsilon$ that satisfies for all $t \geq s \geq 0$

$$P\{B(s,t) \leq A \otimes \mathcal{F}^\varepsilon(s,t)\} \geq 1 - \varepsilon \tag{4}$$

By letting $\varepsilon \longrightarrow 0$ in eq.(3) and eq.(4), we recover the arrival envelopes and service curves of the deterministic calculus with probability one. The statistical network calculus has also been related to other analytical techniques.

The related analytical technique considering statistical multiplexing of arrival flows are well understood for single server under theory of effective bandwidth. An effective bandwidth for an arrival process A with moment generating function

M_A is defined as a non-negative function α such that for all $t \geq s \geq 0$ and $\theta \geq 0$

$$\alpha(\theta, s, t) = \frac{1}{\theta(t-s)} log M_A(\theta, s, t) = \frac{1}{\theta(t-s)} log E[e^{\theta A(s,t)}] \tag{5}$$

The term "effective envelope" as introduced in [5] suggests a connection to the notion of effective bandwidth, but without making that connection explicit. In [2] authors establish a formal relationship between the two concepts, and thus, link the effective bandwidth theory to the statistical network calculus. Given an arrival process A with effective bandwidth α, the effective envelope $\mathcal{G}^\varepsilon$ of the arrival process for any $t \geq s \geq 0$ and $\theta \geq 0$ is given by

$$\mathcal{G}^\varepsilon(s,t) = \inf_\theta \left\{ (t-s)\alpha(\theta, s, t) - \frac{log\varepsilon}{\theta} \right\} \tag{6}$$

In [2] authors use the above relationship to construct an effective envelope for a traffic class if its effective bandwidth is known. Once the effective envelope is fixed at the network ingress, it is used to represent the traffic along the network. This prohibits any consideration of statistical multiplexing along the path of traffic. Moreover the traffic's effective envelope deteriorates along its path by a right-shift of the envelope with the amount (h-1)d [2], where h is number of nodes along the path and d is the a priori delay threshold at each node.

In our approach the traffic's effective bandwidth is used to describe its arrival process in the network along its path. At each network node, effective envelope is constructed from the effective bandwidth of incoming traffic to apply results from the statistical network calculus. This enables efficient utilization of statistical multiplexing of independent traffic flows both at the ingress of the network and along the path of the traffic. The question is how to find the effective bandwidth of the output traffic from a network node? There are two ways: (*i*) a bound for the output effective bandwidth can be found using the results of statistical network calculus with moment generating function as discussed in [17]. But, this bound is conservative due to the use of Boole's inequality. (*ii*) Using *(approximate) invariance* property of the effective bandwidth in the network [10]. This holds already for a surprisingly small number of competing flows even in the presence of aggressive TCP traffic [14]. We use the latter method to find the effective bandwidth of outgoing traffic. In the next section, the invariance of effective bandwidth inside a network is discussed.

3 Invariance of Effective Bandwidth

This section discusses the gap between theoretical invariance results and their application in real networks. In a stable system, the mean rate (ρ) of the aggregated arrival process is always less than the service rate (C) at a node (i.e., $\rho < C$). It is shown in [10] that if the condition $\rho < C$ is strictly valid, the effective bandwidth associated with flows passing a network node does not change, i.e. all flow's effective bandwidth stay the same on their path through the network. The proof in [10] relies on the fact that when there are many independent

sources the queue empties regularly with a high probability. So, how many input processes are needed for this limiting result to be accurate? Numerical simulations suggest that in some cases only a small number of independent inputs are needed to make the input and output look nearly identical. The real question though is how many input processes are needed for reasonable convergence over the scale of interest. We define "Effective Threshold" as the minimum number input processes required at the node to achieve *(approximate) invariance* of the effective bandwidth of output flows. To the best of the author's knowledge, there has been no concrete analytical results to find the effective threshold for a given network model, but a model simulation can be used to find its approximate value.

We analyze convergence of effective bandwidth of the outgoing traffic to that of the incoming traffic using the simulation of a simple network shown in Fig.1. A simple bottleneck with a scalable number of N sources ($S1 \ldots SN$) and N destinations ($D1 \ldots DN$) connected via UDP connections is considered. The bottleneck link (and subsequently router 1, R1) serves at a rate of N·0.2 Mbit/s, i.e. in case of perfect fairness each connection gets a share of 0.2 Mbit/s. we assume a constant data segment size of 128 bytes, i.e. due to IP and UDP headers of 20 bytes and 8 bytes the actual packet size equals 156 bytes. The sources produce exponential on/off traffic constrained by token bucket with parameters (ρ, σ). We change the value of ρ to vary the link utilization (η) of the bottleneck link while maintaining σ as constant with 1024 bytes. We aim to find a sufficiently large number N of connections such that the effective bandwidth of the connections does not change according to the analytical result found in [10] for the many sources limiting regime.

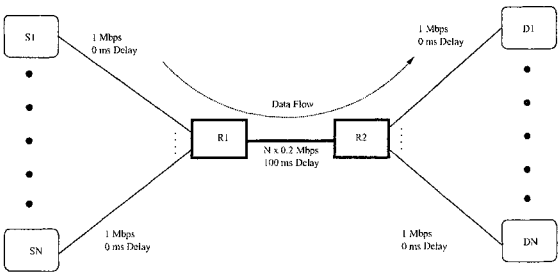


Fig. 1. Scalable bottleneck

We consider the results for the mean bottleneck link load $\eta = 60\%$, i.e., ρ of the token bucket parameter is assigned 122.88kbits/s. The bottleneck link operates at N·0.2 Mbit/s and the buffer of router $R1$ is large enough to avoid any packet losses. The simulation time of each simulation run was set to 600s, i.e. 10 min. real time for each value of N, while N increasing from 1 to 100. The graph in Fig.2 presents the relative discrepancies between the effective bandwidth of

the incoming and the departing (i.e. in and out of R1) flows. In large deviations theory, θ in the effective bandwidth definition eq.(5) is called "space parameter" [11]. It provides the link between the effective bandwidth assigned to a flow and the quality of service, which the flow experiences when passing a node with a service rate equal to its effective bandwidth. The average queue size, average waiting time and the loss probability in the queuing system depend on the space parameter and buffer size, see for instance [12]. The declaration of the QoS demands determines the choice of the space parameter. The choice of space parameter value is in between 0.1 to 1.0 per packet [13, 12]. We study different space parameters $\theta = 0.1$, 0.5, and 1.0 for a time interval of $t = 10$ secs.

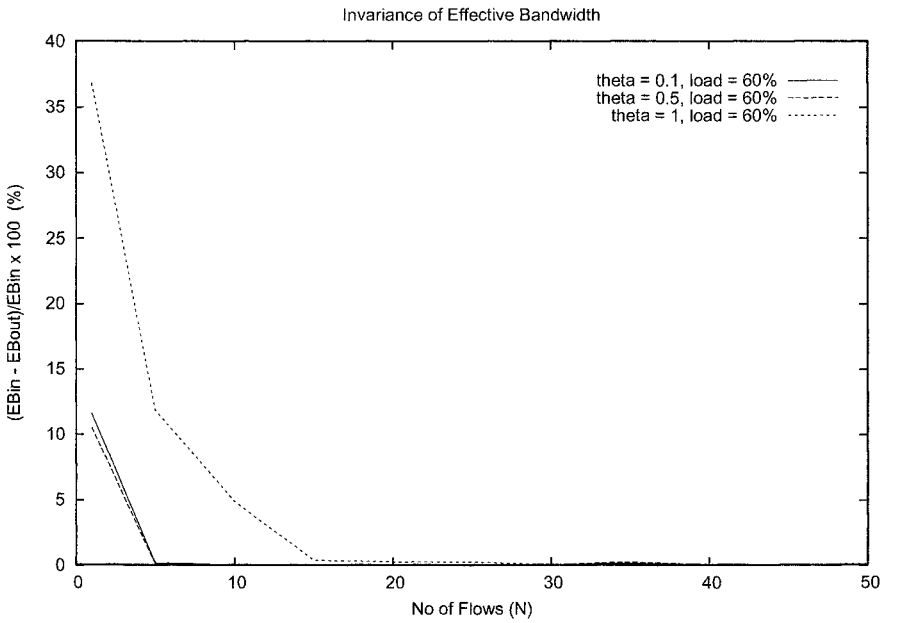


Fig. 2. Relative discrepancies between the effective bandwidths of the incoming and the departing flows for η=60%

It is observed when $\eta = 60\%$, effective bandwidth of the flows stay relatively constant, when passing through the shared router for $N = 6$ flows ($\theta = 0.1$ and 0.5) and $N = 15$ flows ($\theta = 1$). The discrepancies after convergence are between 0.00001 and 0.12 ($\theta = 0.1$), 0 and 0.2 ($\theta = 0.5$), 0.004 and 0.37 ($\theta = 1.0$). The errors increase with increasing space parameter θ, which is a well known effect from effective bandwidth measurements since large θ values boost the exponential functions argument, cf. (5). We have observed that the value of effective threshold depends upon the link utilization η. For example, in a simulation setup when $\eta = 16\%$, the effective threshold is $N = 2$ flows for any $1 \geq \theta \geq 0$ and it increases with increasing η. Apart from the link utilization η and space

parameter θ, the effective threshold depends on the characteristics of the arrival process. To use the *(approximate) invariance* property of effective bandwidth in the network, as a prerequisite, we require the number of independent flows multiplexing at each node is greater than the effective threshold.

4 Improving the Usage of Effective Envelope in the Inner Network

Efficient models for statistical multiplexing are known from the theory of effective bandwidth. Using effective bandwidth to represent the arrival process inside the network within statistical network calculus, allows to inherit some useful large deviation results for an efficient network analysis. The easy handling of effective bandwidth is mostly due to its additivity. For example, in [9], it has been proved that the effective bandwidth of an aggregate flow consisting of different independent flows of different service classes multiplexed together is simply the sum of their independent single flow's effective bandwidth. If $\alpha_x(\theta, s, t)$ and $\alpha_y(\theta, s, t)$ are the effective bandwidth of two independent arrival process $X(s,t)$ and $Y(s,t)$ for all $t \geq s \geq 0$ and $\theta \geq 0$, then the effective bandwidth $\alpha_{x+y}(\theta, s, t)$ of these aggregated flows is given by

$$\alpha_{x+y}(\theta, s, t) = \alpha_x(\theta, s, t) + \alpha_y(\theta, s, t) \tag{7}$$

From *(approximate) invariance* property, the effective bandwidth of a traffic is the same at all points in the network (though the different nodes will typically have different operating points so the values of the function will be different). Hence we can construct effective envelope of the incoming traffic at each network node using the formal relationship established between these two concepts in eq.(6). The constructed effective envelope and effective service curve of the network node are then used to analyze the delay and backlog observed at each node applying the results from statistical network calculus. For the case of an arrival process consisting of an aggregated flow from a large number of independent sources, this formal relationship can be improved by direct application of the Bahadur-Rao Theorem[1] as explained in Lemma 1 in the Appendix. It is important to note that the Bahadur-Rao improvement only affects (decreases) the influence of the violation probability in the formal relationship (eq.6). This helps to construct a tighter effective envelope from the effective bandwidth of the traffic.

The benefit of aggregating flows is that even with a quite small violation probability ε, the statistical arrival curve could have significantly smaller values than the deterministic arrival curve of the aggregated flow. Fig.3 shows the deterministic arrival curve (Det Env), effective envelope (Eff Env) and effective envelope with Bahadur-Rao improvement (Eff Env B-R) of the aggregation of N flows when $\varepsilon = 10^{-9}$. All statistical curves are normalized by dividing by N to be compared with the deterministic arrival curve of a single flow.

Our approach is illustrated in comparison with conventional statistical network calculus approach using a simple example of the tandem nodes with cross

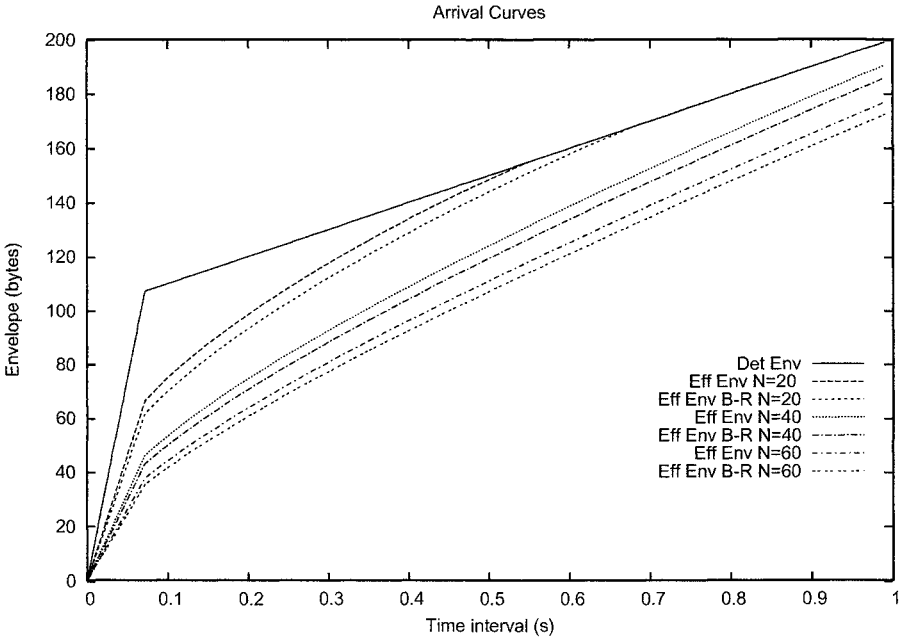


Fig. 3. Deterministic and statistical arrival curves

traffic shown in Fig.4. Let N be the number of through-flows, M_1 and M_2 cross-flows at first node and second node respectively. Note that, in statistical network calculus, the effective envelope is assigned at the network ingress. Once the effective envelope is fixed, no additional statistical multiplexing gain is considered along the path of the flow. Let $\mathcal{G}^{\varepsilon/2}_{N,1}(0,t)$ and $\mathcal{G}^{\varepsilon/2}_{M_1,1}(0,t)$ be the effective envelope of the through-flows and cross-flows at node 1 respectively in interval (0,t] derived using eq.(8), such that the effective envelope of all incoming flows at node 1 is $\mathcal{G}^{\varepsilon}_{N+M_1,1}(0,t) = \mathcal{G}^{\varepsilon/2}_{N,1}(0,t) + \mathcal{G}^{\varepsilon/2}_{M_1,1}(0,t)$. Similarly, let $\mathcal{G}^{\varepsilon/2}_{N,2}(0,t+d)$ and $\mathcal{G}^{\varepsilon/2}_{M_2,2}(0,t)$ be the effective envelopes of the through-flows and cross-flows at node 2 respectively in interval (0,t]. $\mathcal{G}^{\varepsilon}_{N+M_2,2}(0,t) = \mathcal{G}^{\varepsilon/2}_{N,2}(0,t+d) + \mathcal{G}^{\varepsilon/2}_{M_2,2}(0,t)$ is the effective envelope of all incoming flows at node 2, where d is the a priori delay threshold at each node. In our approach, the effective bandwidth is used to describe arrival process inside the network. The basic requirements for the analysis are, (a) the mean arrival rate of the aggregate flow must be strictly less than the service rate of a node, (b) min[N+M_1, N+M_2]$\geq$ effective threshold. The effective envelopes $\overline{\mathcal{G}}^{\varepsilon}_{N+M_1,1}(0,t)$, $\overline{\mathcal{G}}^{\varepsilon}_{N+M_2,2}(0,t)$ are derived from the effective bandwidth of all incoming flows at each node using eq.(8). From eq.(8), it can be seen that $\overline{\mathcal{G}}^{\varepsilon}_{N+M_1,1}(0,t) = \mathcal{G}^{\varepsilon}_{N+M_1,1}(0,t)$ and $\overline{\mathcal{G}}^{\varepsilon}_{N+M_2,2}(0,t) \leq \mathcal{G}^{\varepsilon}_{N+M_2,2}(0,t)$. The latter condition is because the statistical multiplexing gain between the N through-flows and the M_2 cross-flow is efficiently utilized at node 2 using our approach.

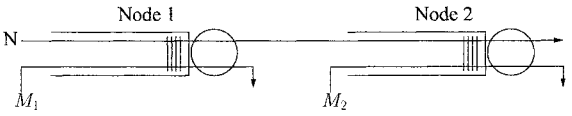


Fig. 4. Tandem nodes with cross-traffic

To show the performance gain of our approach using effective bandwidth to describe the arrival flows within statistical network calculus and additional gain from effective envelope using Bahadur-Rao improvement from eq.(8), we perform an analysis with the same model as in Fig.4. There are $(N + M_1)$ independent incoming flows to node 1, and $M_1 = 15$ (at node 1), $M_2 = 15$ (at node 2) cross-flows and we increase the through-flows N to a maximum feasible value. Every flow has the maximum packet size of 100byte. All flows have the same individual deterministic (σ, ρ, p) arrival curve with $\sigma = 500byte, \rho_r = 200byte/ms, p = 1000byte/ms$. Node 1 and node 2 are both served at $12500kbyte/ms$. Finally, we set the expected delay bound to $d = 2ms$ at every node. The effective threshold for this model was found to be 15 flows using simulation for any $1 \geq \theta \geq 0$. Fig.5 shows the maximum feasible soft real-time through-flows N calculated, using the derived effective envelopes and statistical network calculus results found in [2], for the demo system under different violation probabilities ε. It should be noted that the difference between the two approaches is seen only at node 2.

In summary, from Fig.5 it is observed that using the arrival curves from network calculus (Det Env) enables 22 through-flows, the effective envelope from eq.(6) (Eff Env) enables 30 to 36 through-flows, and effective envelope using Bahadur-Rao improvement from eq.(8) (Eff Env B-R) enables 31 to 38 through-flows repectively. Clearly, Bahadur-Rao improvement provides a better performance than the existing approaches. If *(approximate) invariance* property of effective bandwidth is considered, effective envelope (Inv Eff Env) and effective envelope (Inv Eff Env B-R) using Bahadur-Rao improvement enable 47 to 57 and 49 to 60 through-flows respectively. It can be seen that improvement is achieved by exploiting the statistical multiplexing between through-flows and cross-flows at both nodes, whereas, the conventional approach considers the statistical multiplexing gain only at the ingress node.

5 Conclusion

In this paper, we have explored the invariance condition of the effective bandwidth along the feedforward network. We used the *(approximate) invariance* property and the relationship between effective bandwidth and effective envelope to make an efficient network analysis using statistical network calculus. We proposed an improvement to the formal relationship [2] between effective envelope and effective bandwidth using Bahadur-Rao theorem [1] to derive a tighter bound for the arrival process in the network. This results in an efficient

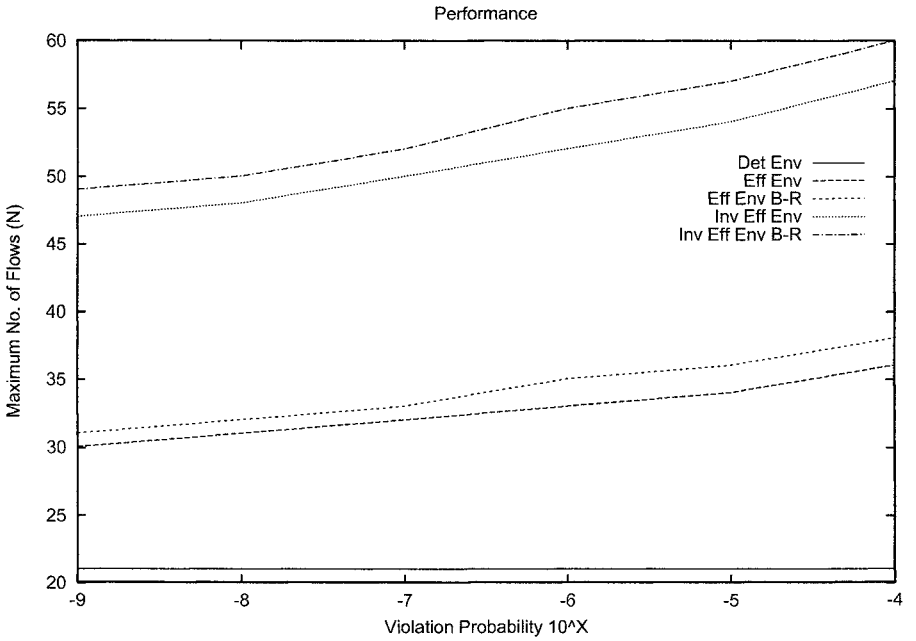


Fig. 5. The performance from scenario 1 ($M_1 = 15$, $M_2 = 15$)

use of the network resources and an increase in the number of traffic flows that can be accommodated in the network without violating their soft real-time QoS constraints.

References

1. Bahadur, R., R. and Ranga Rao, R.,: On deviations of the sample mean. Ann. Math. Statist. 31 1960.
2. Li, C., Burchard, A., and Liebeherr, J.,: A Network Calculus with Effective Bandwidth. Technical Report CS-2003-20, University of Virginia, 2003.
3. Chang, C.-S.,: Performance Guarantees in Communication Networks. Springer-Verlag, 2000.
4. Le Boudec J.-Y., and Thiran P.,: Network Calculus A Theory of Deterministic Queuing Systems for the Internet, ser. LNCS. Springer-Verlag, 2001, no. 2050.
5. Boorstyn, R.-R., Burchard, A., Liebeherr, J., and Oottamakorn C.,: Statistical Service Assurances for Traffic Scheduling Algorithms, IEEE Journal on Selected Areas in Communications, 18(12):2651-2664, 2000.
6. Burchard, A., Liebeherr, J., and Patek, S. D.,: A calculus for end-to-end statistical service guarantees (revised). Technical Report CS-2001-19, University of Virginia, Computer Science Department, May 2002.
7. Likhanov, N., and Mazumdar, R. R.,: Cell loss asymptotics for buffers fed with a large number of independent stationary sources, Journal of Applied Probability, 1999.

8. Montgomery, M., and de Veciana, G.,: On the relevance of time scales in performance oriented traffic characterizations. In Proc. of IEEE INFOCOM'96, pp. 513-520, April 1996.
9. Chang, C.-S., and Thomas, J.A.,: Effective bandwidth in high speed digital networks, IEEE Journal on Selected Areas in Communications, vol. 13, no. 6, 1995.
10. Wischik, D.,: The output of a switch, or, effective bandwidths for networks, Queueing Systems, Volume 32, 1999.
11. Kelly,F. P.,: Notes on effective bandwidths, ser. Royal Statistical Society Lecture Notes. Oxford University, 1996, no. 4.
12. Yang, J., and Devetsikiotis, M.,: On-line estimation, network design and performance analysis, Proceedings of International Teletraffic Congress - ITC 17, Elsevier Science. 2001
13. Tartarelli, S., Falkner, M., Devetsikiotis, M., Lambadaris, I., and Giordano,S.,: Empirical effective bandwidths, Proc. of IEEE GLOBECOM 2000, vol. 1, 2000.
14. Abendroth, D., and Killat, U.,: An advanced traffic engineering approach based on the approximate invariance of effective bandwidths, Telecommunication Systems Journal, Kluwer Publishers, 2004, vol 27 (2-4).
15. Jiang, Y. 2006. A basic stochastic network calculus. SIGCOMM Comput. Commun. Rev. 36, 4 (Aug. 2006), 123-134.
16. Ciucu, F., Burchard, A., and Liebeherr, J.,: A network service curve approach for the stochastic analysis of networks, in Proc. ACM SIGMETRICS, June 2005, 279-290.
17. Fidler, M.,: An End-to-End Probabilistic Network Calculus with Moment Generating Functions. Proceedings of IWQoS, June 2006.

Appendix

Lemma 1 *Let arrival process A be the superposition of N independent flows of J types, each having an effective bandwidth α_j. Let $n = (n_1, n_2, \ldots, n_J)$ define the traffic proportion, where $N \cdot n_j$ be the number of flows of type j. Then the effective envelope $\mathcal{G}^\varepsilon$ and effective bandwidth α of the aggregate flow will have the following relationship for any $t \geq s \geq 0$ and $\theta \geq 0$:*

$$\mathcal{G}^\varepsilon(s,t) = \inf_\theta \left\{ (t-s)\alpha(\theta,s,t) - \frac{log\varepsilon'}{\theta} \right\} \tag{8}$$

where

$$log\varepsilon' = log\varepsilon + \frac{\frac{1}{2}log(-4\pi log\varepsilon)}{1 - \frac{1}{2log\varepsilon}} \tag{9}$$

$$\alpha(\theta,s,t) = \sum_{j=1}^{J} N n_j \alpha_j(\theta,s,t) \quad (from\ eq.(7)) \tag{10}$$

Proof: To prove the statement, fix t ≥ s ≥ 0. By the the Bahadur-Rao Theorem[1], we have for any x(s,t)

$$P\{A(s,t) \geq x(s,t)\} \approx \frac{e^{-NI(x(s,t)/N)}}{\theta\sqrt{2\pi\sigma^2 N}} \left(1 + o\left(\frac{1}{N}\right)\right) \tag{11}$$

$$\approx e^{-NI(x(s.t)/N) - \frac{1}{2}log(2\pi N\sigma^2\theta^2)} \tag{12}$$

where,

$$I(x(s,t)/N) = \theta \frac{x(s,t)}{N} - (t-s)\theta \sum_{j=1}^{J} n_j \alpha_j(\theta, s, t)$$

and θ is the unique solution of the equation,

$$\frac{M^{'}{}_{A/N}(\theta, s, t)}{M_{A/N}(\theta, s, t)} = \frac{x(s,t)}{N}$$

and,

$$\sigma^2 = \left(\frac{M^{'}{}_{A/N}(\theta, s, t)}{M_{A/N}(\theta, s, t)} - \left(\frac{M^{''}_{A/N}(\theta, s, t)}{M_{A/N}(\theta, s, t)} \right)^2 \right)$$

In large deviations theory, $I(x(s,t)/N)$ is referred to as the rate function.

The term $\frac{1}{2}log(2\pi N\sigma^2\theta^2)$ can be approximated by $\frac{1}{2}log(4\pi NI(x(s,t)/N))$ [8]. If the violation probability of the arrival process is $\varepsilon = e^{-\gamma}$, then by equating eq.(12) to ε, we have

$$-NI(x(s,t)/N) - \frac{1}{2}log(2\pi N\sigma^2\theta^2) = -\gamma \Leftrightarrow NI(x(s,t)/N) = \gamma - \frac{1}{2}log(2\pi N\sigma^2\theta^2)$$

Substituting $\frac{1}{2}log(2\pi N\sigma^2\theta^2)$ by $\frac{1}{2}log(4\pi NI(x(s,t)/N))$

$$NI(x(s,t)/N) = \gamma - \frac{1}{2}log(4\pi NI(x(s,t)/N))$$

Setting $NI(x(s,t)/N) = \gamma + \epsilon$ in the last equation and taking the expansion of the logarithm on the right-hand side, i.e., $log(4\pi NI(x(s,t)/N)) = log(4\pi(\gamma + \epsilon)) \approx log(4\pi\gamma) + \frac{\epsilon}{\gamma}$, we obtain

$$\begin{aligned} \gamma + \epsilon &\approx \gamma - \frac{1}{2}log(4\pi\gamma) - \frac{1}{2\gamma}\epsilon \\ \Rightarrow \left(1 + \frac{1}{2\gamma}\right)\epsilon &\approx -\frac{1}{2}log(4\pi\gamma) \\ \Rightarrow \epsilon &\approx -\frac{\frac{1}{2}log(4\pi\gamma)}{1 + \frac{1}{2\gamma}} \end{aligned}$$

Substituting the last equation in $NI(x(s,t)/N) = \gamma + \epsilon$ gives

$$NI(x(s,t)/N) \approx \gamma - \frac{\frac{1}{2}log(4\pi\gamma)}{1 + \frac{1}{2\gamma}} = -\left(log\varepsilon + \frac{\frac{1}{2}log(-4\pi log\varepsilon)}{1 - \frac{1}{2log\varepsilon}} \right) = -log\varepsilon^{'}$$

Solving the above equation for $x(s,t)$, proves the claim.

Comparison of Preemptive and Preserving Admission Control for the UMTS Enhanced Uplink

Andreas Mäder und Dirk Staehle

University of Würzburg, Department of Distributed Systems
Am Hubland, D-97074 Würzburg, Germany
{maeder, staehle}@informatik.uni-wuerzburg.de

Abstract The UMTS enhanced uplink provides high bit rate radio bearers with fast rate control for packet switched radio traffic. Resource Managament in UMTS networks with the enhanced uplink has to consider the requirements of the dedidated channel users and the enhanced uplink users on the shared resource, i.e. the cell load. We propose an analytical model for such a system and evaluate the impact of two resource management strategies, one with preemption for dedicated channels and one without, on key QoS-indicators like blocking and dropping probabilities as well as user and cell throughput.
Keywords: WCDMA, UMTS, Enhanced Uplink, HSUPA, radio resource management, radio network planning

1 Introduction and Related Work

The enhanced uplink (sometimes also referred to as high speed uplink packet access - HSUPA) marks the next step in the evolution process of the UMTS. Introduced with UMTS release 6 and specifically designed for the transport of packet switched data, it promises higher throughput, reduced packet delay and a more efficient radio resource utilization. A detailed overview can be found e.g. in [1] or [2]. The enhanced uplink introduces a new transport channel, the Enhanced-DCH (E-DCH) and three new signaling channels. The E-DCH can be seen as an "packet-optimized" version of the DCH. The major new features are: Hybrid ARQ, implemented similarly as in the high speed downlink packet access (HSDPA), NodeB-controlled fast scheduling, and reduced transport time intervals (TTI) of 2 ms. In a UMTS network with enhanced uplink it is expected that QoS users will use the normal dedicated channels (DCH) while best-effort users will use the new enhanced dedicated channel (E-DCH).

In [3] we propose an analytical model and a radio resource management strategy based on the notion of effective bandwidths for the UMTS enhanced uplink with a preserving admission control strategy, meaning that DCH and E-DCH connections have equal priority on call arrival. We also showed the impact of one-by-one and parallel scheduling on the system performance. The focus of this work is the comparison of admission control strategies with and without

priority between DCH and E-DCH connections. The admission control without priority is called *preserving*, while the admission control with priority of DCH connections is call *preemptive* since here E-DCH connections may be dropped for the sake of DCH connections.

The analytical model considers packet data streams as a *flow*, i.e. it is seen as a continuous stream of data regardless of the underlying protocol. Related work which can be found in the literature is e.g. [4], where a queueing analysis for the CDMA uplink with best-effort services is presented. A similar approach has been taken in [5], which introduces a dynamic slow down approach for the best-effort users. Preemption for QoS-users is e.g. considered in [6] for a GPRS/HSCSD system.

The question is whether the concept of preemption is a suitable way to increase the service quality for QoS-users without degrading the service for best-effort users too strong. The answer to this question is, however, always in the hand of the operator who defines acceptable service qualities for its customers. We provide an analytical tool to calculate the blocking and dropping probabilities as well as the user bit rates of the enhanced uplink users.

The rest of this paper is organized as follows: In Sec. 2 we define the radio resource management strategy which provides the frame for our calculations. This forms the base for the interference and cell load model in Sec. 3. In Sec. 4, we describe the E-DCH rate assignment and in Sec. 5 the admission control mechanism, which is then used for a queueing model approach in Sec. 6. In Sec. 7, we show some numerical examples and finally we conclude the paper with Sec. 8.

2 Radio Resource Management for the E-DCH Best Effort Service

Radio resource management (RRM) for the E-DCH users is primarily done in the NodeBs, which control the maximum transmit power of the mobiles and therefore also the maximum user bit rate. The NodeBs send scheduling grants on the absolute or relative grant channel (AGCH and RGCH, resp.), which either set the transmit power to an absolute value or relative to the current value. The mobiles then choose the transport block size (TBS) which is most suitable to the current traffic situation and which does not exceed the maximum transmit power. The grants can be sent every TTI, i.e. every 2 ms, which enables a very fast reaction to changes of the traffic or radio conditions. Grants can be received from the serving NodeB and from non-serving NodeBs. However the latter may just send relative DOWN grants to reduce the other-cell interference in their cells. In our model, we consider grants from the serving NodeB only.

Generally, the WCDMA uplink is interference limited. Therefore, following [7], we define the load in a cell as

$$\hat{\eta} = \frac{\hat{I}_D + \hat{I}_E + \hat{I}_{oc}}{\hat{I}_0 + W\hat{N}_0}, \tag{1}$$

with $\hat{I}_D$ and $\hat{I}_E$ as received powers from the DCH and E-DCH users[1] within the cell, $\hat{I}_{oc}$ as other-cell interference from mobiles in adjacent cells, W as system chip rate, $\hat{N}_0$ as thermal noise power spectral density and $\hat{I}_0 = \hat{I}_D + \hat{I}_E + \hat{I}_{oc}$. It can be readily seen that this load definition allows the decomposition of the cell load after its origin, hence we define

$$\begin{aligned} \hat{\eta} &= \frac{\hat{I}_D}{\hat{I}_0 + W\hat{N}_0} + \frac{\hat{I}_E}{\hat{I}_0 + W\hat{N}_0} + \frac{\hat{I}_{oc}}{\hat{I}_0 + W\hat{N}_0} \\ &= \hat{\eta}_D + \hat{\eta}_E + \hat{\eta}_{oc} \end{aligned} \tag{2}$$

subject to $\hat{\eta} < 1$. The goal of the RRM is now twofold: First, the cell load should be below a certain maximum load in order to prevent outage. Second, the RRM tries to maximize the resource utilization in the cell to provide high service qualities to the users. The second goal allows also the interpretation of the maximum load as a target load, which should be met as close as possible. Since the DCH-load and the other-cell load cannot be influenced in a satisfying way, the E-DCH load can be used as a means to reach the target cell load. The fast scheduling gives operators the means to use the E-DCH best-effort users for "water-filling" the cell[2] load at the NodeBs up to a desired target. This radio resource management strategy is illustrated in Fig. 1. The total cell load comprises the varying other-cell load, the load generated by DCH users and the E-DCH load. The received power for the E-DCH users is adapted such that the total cell load is close to the maximum load. However, due to the power control error and the other-cell interference there is always the possibility of a load "overshoot". The probability for such an event should be kept low. So, the cell

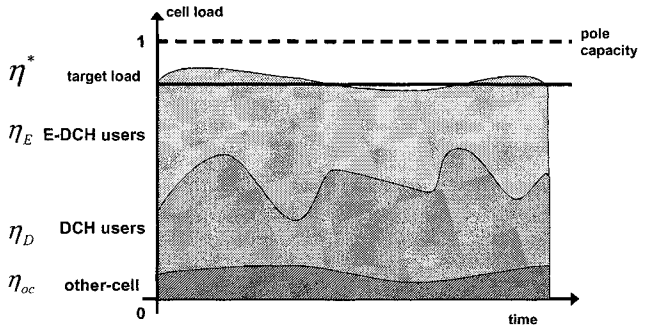


Figure 1. Illustration of the RRM principles for the E-DCH best-effort service.

load is a random variable due to fast fluctuation of the received E_b/N_0 values. We define that the goal of the RRM is to keep the probability of the total cell

[1] Note that variables $\hat{x}$ are in linear and x are in dB scale

[2] corresponding to a sector in case of multiple sectors per NodeB

load below a maximum tolerable probability p_t:

$$P\{\hat{\eta} \geq \hat{\eta}^*\} \leq p_t. \tag{3}$$

This means that the received signal power (i.e. the E-DCH interference) of the E-DCH users depends on the amount of dedicated channel and other-cell interference. More precisely, the E-DCH users are slowed down if the DCH or the other-cell load is growing, or are speed up, if more radio resources are available for the E-DCH users. If we now assume that the buffers in the mobiles of the E-DCH users are always saturated, we can use this relation to calculate the grade-of-service the E-DCH users receive depending on the scheduling strategy.

3 Interference and Load Model

Let us consider a NodeB in a UMTS network serving a single sector or cell, respectively. In the cell is a number of DCH users, each connected with a service class $s \in \mathcal{S}$. The service classes are defined by bitrate and target-E_b/N_0-value. Additionally, n_E E-DCH users are in the system. The state vector $\bar{n}$ comprises the users per DCH service class, n_s, and the E-DCH users n_E:

$$\bar{n} = (n_1, \ldots, n_{|S|}, n_E). \tag{4}$$

Each mobile power controlled by the NodeB perceives an energy-per-bit-to-noise ratio (E_b/N_0), which is given by

$$\hat{\varepsilon}_k = \frac{W}{R_k} \frac{\hat{S}_k}{W\hat{N}_0 + \hat{I}_0 - \hat{S}_k}. \tag{5}$$

In this equation, W is the chip rate of 3.84Mcps, R_k is the radio bearer information bit rate, $\hat{N}_0$ is the thermal noise power density, $\hat{S}_k$ is the received power of mobile k and $\hat{I}_0$ is the multiple-access interference (MAI) including the own- and other-cell interference. We assume imperfect power control, so the received E_b/N_0 is a lognormally distributed r.v. with the target-E_b/N_0-value ε_k^* as mean value [8] and parameters $\mu = \varepsilon_k^* \cdot \frac{\ln(10)}{10}$ and $\sigma = \mathrm{Std}[\varepsilon_k] \cdot \frac{\ln(10)}{10}$. The received power of each mobile is calculated from (5) as

$$\hat{S}_k = \hat{\omega}_k \cdot (W\hat{N}_0 + \hat{I}_0) \quad \text{with} \quad \hat{\omega}_k = \frac{\hat{\varepsilon}_k R_k}{W + \hat{\varepsilon}_k R_k}. \tag{6}$$

We define the r.v. ω_k as *service load factor* (SLF) depending on the bit rate and the E_b/N_0-value. The sum of all concurrently received powers constitutes the received own-cell interference, i.e.

$$\hat{I}_D(\bar{n}) = \sum_{s \in \mathcal{S}} \sum_{k \in n_s} \hat{S}_k \quad \text{and} \quad \hat{I}_E(\bar{n}) = \sum_{j \in n_E^a} \hat{S}_j. \tag{7}$$

$\hat{I}_D$ is the total received power of the DCH users and $\hat{I}_E$ of the E-DCH users. The substitution of $\hat{I}_D$ and $\hat{I}_E$ in Eq. (2) with Eq. (7) gives us then the load definitions depending on $\bar{n}$:

$$\hat{\eta}_D(\bar{n}) = \sum_{s \in \mathcal{S}} \sum_{k \in n_s} \hat{\omega}_k \quad \text{and} \quad \hat{\eta}_E(\bar{n}) = \sum_{j \in n_E} \hat{\omega}_j, \tag{8}$$

and the total load as

$$\hat{\eta}(\bar{n}) = \hat{\eta}_D(\bar{n}) + \hat{\eta}_E(\bar{n}) + \hat{\eta}_{oc}. \tag{9}$$

We assume the service load factors as lognormal r.v.'s with parameters μ, σ derived from the mean and variance of the E_b/N_0 distributions. These parameters depend on the service class of the users, but are equal for all users within one class. So we can write $E[\hat{\omega}_k] = E[\hat{\omega}_s]$ for all mobiles k with the same service class s. The other-cell load η_{oc} is modeled as a lognormal r.v. with constant mean and variance.

Since the total load $\hat{\eta}$ is a sum of independent lognormally distributed r.v.'s, we assume that $\hat{\eta}$ also follows a lognormal distribution [9]. We get the distribution parameters from the first moment and variance of the cell load which can be calculated directly from the moments of the SLFs:

$$E[\hat{\eta}(\bar{n})] = \sum\nolimits_{s \in \mathcal{S}} n_s \cdot E[\hat{\omega}_s] + n_E^a \cdot E[\hat{\omega}_E] + E[\hat{\eta}_{oc}]. \tag{10}$$

The variance is calculated analogously. The accuracy of this approach is validated e.g. in [10]. Another novelty of the E-DCH is Hybrid ARQ (HARQ), which combines the automatic-repeat-request protocol with code combining techniques. The effect of HARQ can be modeled as an constant gain which is included in the target-E_b/N_0 of the E-DCH and with an additional overhead on the mean data volumes of the E-DCH.

4 Rate Assignement

The available E-DCH load depends on the DCH and other-cell load. The task of the RRM is to assign each E-DCH mobile a service load factor ω such that the E-DCH load is completely utilized if possible. Generally, the user bit rate depends on the E-DCH cell load which may be generated without violating the RRM target in (3). The channel bit rate of the E-DCH is defined by the amount of information bits which can be transported within one TTI. This quantity is defined in [11] by the set of transport block sizes $\mathcal{TBS}$. With a TTI of 2ms, the information bit rate per second follows as $R_{i,E} = TBS_i \cdot 1/\text{TTI}$, where $i = 1, \ldots, |\mathcal{TBS}|$ indicates the index of the TBS. We further define $R_{0,E} = 0$. With this interpretation we can map the E-DCH bit rate to a service load factor according to Eq. (6) as

$$\hat{\omega}_{i,E} = \frac{\hat{\varepsilon}_E R_{i,E}}{\hat{\varepsilon}_E R_{i,E} + W}, \tag{11}$$

where $\hat{\varepsilon}_E$ is the E_b/N_0 for the E-DCH RB. Note that here we assume that the target-E_b/N_0-values are equal for all rates. However, this restriction can be easily

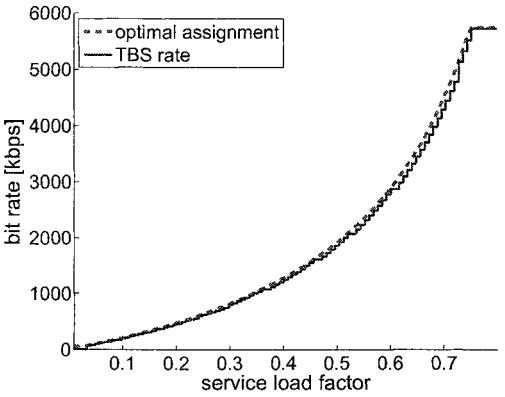


Figure 2. Mapping of service load factors to bit rates

avoided by introducing individual target-E_b/N_0-values for each rate (and ω), if they are available. The next step is to select the information bit rate such that (3) is fulfilled:

$$R_E(\bar{n}) = \max\{R_{i,E} | P(\hat{\eta}_D(\bar{n}) + n_E \cdot \hat{\omega}_{i,E} + \eta_{oc} \geq \hat{\eta}^*) \leq p_t\} \tag{12}$$

Figure 2 shows the mapping of the service load factors to information bit rates in case of a target-E_b/N_0 of 3 dB. The optimal case indicated by the dashed line is calculated from the definition of the service load factors as $R_{\text{opt}} = \frac{\omega \cdot W}{\hat{\varepsilon}^* (1-\omega)}$. The solid line shows the corresponding rate calculated from the TBS. Both curves are very close to each other, and we see that for high SLFs, a small change means a large change on the bit rate. The non-linear dependency between bit rate and SLF is the basis for the argument that a slow-down (in terms of bit rate) of the users leads to an increased system capacity in terms of admissible sessions if an admission control based on the cell load is used ([4] and [5]). However, if we define capacity as the cumulated bit rate per cell, the capacity shrinks with the number of parallel transmitting users due to the increased interference. This is the argument for the throughput gain with one-by-one scheduling in [3].

5 Admission Control

The admission control (AC) is responsible for keeping the cell load below the maximum load. Generally, we model the AC on basis of the RRM target condition. We distinguish between two RRM policies for incoming QoS users: The first, which we call *preserving* treats E-DCH and QoS equally, which means that an incoming connection of either class is blocked if there are not enough resources available. The second, which we call *preemptive*, gives priority to QoS users, which means that eventually active best effort connections may be dropped from the system in order to make room for the incoming QoS user. In both policies

existing E-DCH connections are slowed-down if the number of QoS-connections increases. However, with the preserving strategy incoming QoS-calls are blocked if the RRM cannot slow-down the E-DCH connections any more. With the preemptive strategy, one or more E-DCH connections are dropped from the system in this case, meaning that blocking for the QoS users occurs only if nearly all resources are occupied by QoS connetions, cf. Fig. 3.

If a new connection is to be established to the network, the AC is done in two steps: At first, the amount of resources ω which the incoming connection will occupy is identified. In case of a QoS-connection, this is simply ω_s. In case of an E-DCH connection, incoming connections are admitted if a minimum bit rate $R_{min,E}$ can be guaranteed. The corresponding SLF is denoted with $\omega_{min,E}$. Let us further denote with $\bar{n}^+$ the state vector $\bar{n}$ plus the incoming connection with service class s or with an additional E-DCH connection. The second step is then to estimate the probability for exceeding the maximum load with the new connection included. This step depends on the implemented policy:

Preserving Policy: In the preserving case, we calculate the parameters for the distribution of the expected cell load η_{AC} as in (10), but with $\omega_{\min,E}$ for the E-DCH users:

$$\eta_{AC}(\bar{n}^+) = \eta_D(\bar{n}^+) + n_E^+ \cdot \hat{\omega}_{\min,E} + \eta_{oc}, \tag{13}$$

where n_E^+ is the number of E-DCH mobiles with the incoming mobile included, if any. So, if the probability $P(\eta_{AC} \geq \eta^*)$ is higher than the target probability p_t, the connection is rejected, otherwise the connection is admitted.

Preemptive Policy: With preemption, the incoming call is admitted if enough resources are available such that $P(\eta_{AC} \geq \eta^*) \leq p_t$, as in the preserving case. However, if the resources are insufficient, we distinguish two cases: If the incoming call belongs to an E-DCH user, the call is blocked. If the incoming call belongs a QoS user, the RRM calculates from the service requirement ω_s the number of E-DCH connections with minimum rate $R_{E,min}$ which must be dropped from the system such that the incoming call can be admitted. The number of E-DCH connections $n_d(\bar{n}, s)$ which must be dropped depends on the current state and on the SLF of the incoming QoS-connection. It is given by the following rule:

$$n_d(\bar{n}, s) = \min\{n | P(\eta_D(\bar{n}^+) + (n_E - n) \cdot \omega_{\min,E} + \eta_{oc} \geq \eta^*) \leq p_t\}. \tag{14}$$

Note that $0 \leq n_d \leq \lceil \frac{\omega_s}{\omega_{\min,E}} \rceil$. Blocking for QoS-users occurs if the number of E-DCH connections is too low to meet the requirements of the service class, i.e. if $n_d(\bar{n}, s) > n_E$. Blocking for E-DCH users occurs if the existing connections cannot be slowed down any further, due to the constraint on the minimum bit rate.

After admission control, the RRM executes the rate assignment as in Eq. (12) to adjust the bit rate of the E-DCH users to the new situation. Figure 4 illustrates the principle of admission control and rate selection. It shows the mean and the $(1 - p_t)$-quantile (here $p_t = 5\%$) of the cell load distribution for 5 DCH users and an increasing number of E-DCH users. The target load is

$\hat{\eta}^* = 0.85$. Due to the discretization of the available rates, the $(1 - p_t)$-quantile does not exactly meet the target-load, but stays just below. Since the coefficient of variation of the cell load is decreasing with the number of users in the system, the mean load comes closer to the target load with an increasing number of E-DCH users.

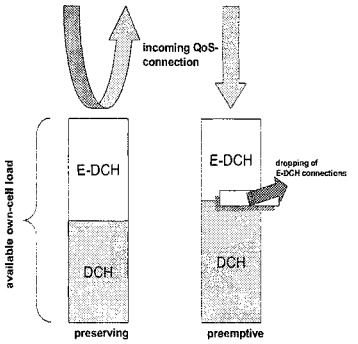


Figure 3. Principle of the preserving and preemptive policy.

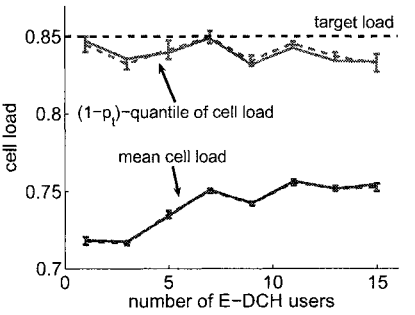


Figure 4. Mean cell load and 95%-quantiles.

6 Performance Evaluation

Now we assume that all calls arrive with exponentially distributed interarrival times with mean $\frac{1}{\lambda}$. The users choose a DCH service class or the E-DCH with probability $p_{s|E}$, hence the arrival rates per class are $\lambda_{s|E} = p_{s|E} \cdot \lambda$. The holding times for the DCH calls are also exponentially distributed with mean $\frac{1}{\mu_s}$. For the E-DCH users we assume a volume based user traffic model [12]. With exponentially distributed data volumes, the state-dependent departure rates of the E-DCH users are then given by

$$\mu_E(\bar{n}) = n_E \cdot \frac{R_E(\bar{n})}{E[V_E]}, \tag{15}$$

where $E[V_E]$ is the mean traffic volume of the E-DCH users.

The resulting system is a multi-service $M/M/n - 0$ loss system with state dependent departure rates for the E-DCH users. We are now interested in calculating the steady-state distribution of the number of users in the system. Since the joint Markov process is not time-reversible which can be instantly verified with Kolomogorov's reversibility criterion, no product form solution exists. The steady state probabilities follow by solving

$$Q \cdot \bar{\pi} = 0 \quad \text{s.t.} \quad \sum \pi = 1 \tag{16}$$

for $\bar{\pi}$, where Q is the transition rate matrix. The rate matrix Q is defined with help of the bijective index function $\phi(\bar{n}) : \Omega \rightarrow N$, which maps the state vector $\bar{n}$ to a single index number. The transition rate $q(\phi(\bar{n}), \phi(\bar{n} \pm \bar{1}))$ in the rate matrix between states $\bar{n}$ and $\bar{n} \pm \bar{1}$ is then

$$q(\phi(\bar{n}), \phi(\bar{n} + \bar{1}_s)) = \lambda_s \tag{17}$$

$$q(\phi(\bar{n}), \phi(\bar{n} + \bar{1}_E)) = \lambda_E \tag{18}$$

$$q(\phi(\bar{n}), \phi(\bar{n} - \bar{1}_s)) = n_s \cdot \mu_s \tag{19}$$

$$q(\phi(\bar{n}), \phi(\bar{n} - \bar{1}_E)) = \mu_E(\bar{n}) \tag{20}$$

for all valid states in the state space Ω and $q(\phi(\bar{n}), \phi(\bar{n} \pm \bar{1})) = 0$ otherwise. The sets of $\Omega^+_{ps,b}$ states where blocking occurs in the preserving case are defined by the condition $P(\eta(\bar{n}^+) \geq \eta^*) > p_t$, i.e. they form the 'edges' of the state space. With preemption, an E-DCH connection is dropped if $P(\eta(\bar{n}^{+s}) \geq \eta^*) > p_t$ and $n_d(\bar{n}, s) \geq \lceil \frac{\omega_s}{\omega_{\min,E}} \rceil$, i.e. in case of an incoming QoS connection. We define this set as $\Omega^{+s}_{pe,d}$. Blocking occurs then in the set $\Omega^{+s}_{pe,b} = \Omega^{+s}_{ps,b} \setminus \Omega^{+s}_{pe,d}$. The set of blocking states for E-DCH connections is the same for both policies. For the preemptive policy, an additional entry in the transition rate matrix is generated for states where preemption may occur:

$$q(\phi(\bar{n}), \phi(\bar{n} + \bar{1}_s - \bar{n}_d(\bar{n}, s))) = \lambda_s. \tag{21}$$

As performance measures we choose the service-dependent call blocking probabilities P_s, the call dropping probability P_d which applies only in the case of the preemptive strategy and the mean user bit rate $E[R_U]$ achieved by the E-DCH users. The call blocking probabilities are easily calculated as the sum of all states probabilities in which blocking may occur:

$$P_s = \sum_{\bar{n} | \bar{n} \in \Omega^{+s}_b} \pi(\bar{n}). \tag{22}$$

Note that we omit the qualifier for the admission control policy. We define the call dropping probability in our analysis as the probability that an E-DCH connection is dropped if a QoS-call is arriving in the system. This probability is given by

$$P_d = \sum_{\bar{n} | \Omega^{+s}_{pe,d}} \frac{\pi(\bar{n}) \cdot \sum_{s' \in \mathcal{S}} P^a_{s'} \cdot P^{\text{sel}}_{s'}}{\sum_{\bar{n}' | n_E > 0} \pi(\bar{n}')}, \tag{23}$$

where P^a_s is the probability that the incoming connection is of class s and P^{sel}_s is the probability that an active E-DCH connection is selected for dropping:

$$P^a_s = \frac{\lambda_s}{\sum_{s' \in \mathcal{S}} \lambda_{s'}}, \quad \text{and} \quad P^{\text{sel}}_s = \frac{n_d(\bar{n}, s)}{n_E}. \tag{24}$$

We define further the mean throughput per user at a random time instance as

$$E[R_U] = \sum_{R_E > 0} R_E \cdot \frac{\sum_{\bar{n} | R_E(\bar{n}) = R_E} n_E \cdot \pi(\bar{n})}{\sum_{\bar{n}' | n_E > 0} n'_E \cdot \pi(\bar{n}')}, \tag{25}$$

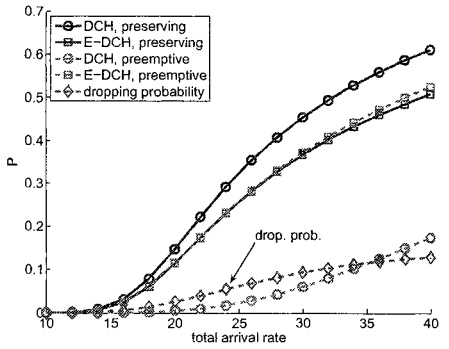


Figure 5. Blocking and dropping probabilities for QoS and E-DCH users

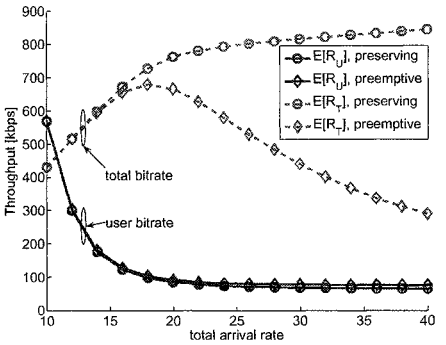


Figure 6. Mean user and cell bit rates.

which is conditioned with the probability that at least one E-DCH user is in the system, because otherwise the mean does not exist.

7 Numerical Results

In this section we give some numerical examples for our model. Our scenarios, if not stated otherwise, consist of two service classes: 64 kbps QoS-users (i.e. DCH users) with a target-E_b/N_0 of 4 dB and the E-DCH best effort users with a target-E_b/N_0 of 3 dB. The service probabilities are $p_1 = 0.4$ and $p_E = 0.6$. In Fig. 5, blocking and dropping probabilities for both admission policies are shown. The curves with a circle marker indicate the blocking probabilities for the 64 kbps QoS users, while the curves with a square marker show the blocking probabilities for the E-DCH users. The dashed line with diamond markers shows the dropping probabilities in case of preemption. Although a system with such high blocking probabilities would be considered as heavily overloaded, we show these results for a better understanding of the effect of preemption. It can be stated that preemption leads to an enormous performance gain for the QoS users, which is caused by the substantially smaller sets of states where blocking can occur at all. The blocking probabilities for the E-DCH users, however, are nearly identical and only begin to differ from each other under very high load. The dropping probabilities do not exceed approx. 10% because in high load regions the system is nearly fully occupied by QoS users.

The impact of preemption on the user and cell bit rates (defined as the cumulated bit rates of all users at any time) is shown in Fig. 6. The user throughputs have solid lines, while the cell throughputs have dashed lines. The expected user throughputs in both cases $E[R_U]$ are nearly identical with a slight advantage for the preemptive case. However, due to the dropping of users the total cell throughput $E[R_T]$ in the preemptive case is significantly lower than in the preserving case. Since the cell throughputs also consider the case if no E-DCH user at all is in the system, the curves are first increasing and then decreasing. In

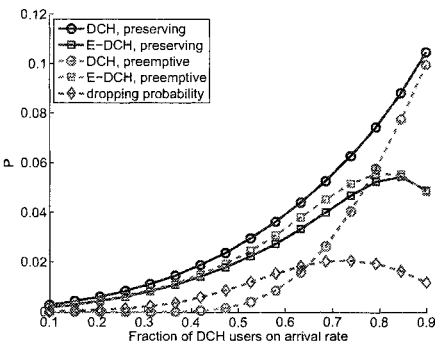


Figure 7. Impact of preemption depends on ratio between DCH and E-DCH users.

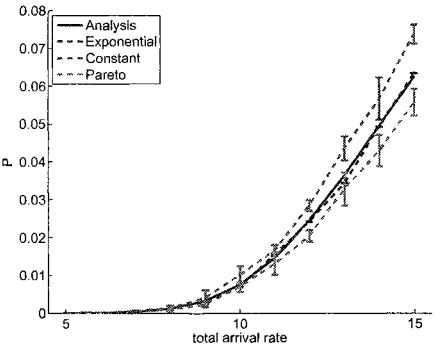


Figure 8. Sensitivity of the dropping probabilities against volume size distribution.

the next scenario we fix the total arrival rate to 15 and vary the ratio between DCH and E-DCH arrivals from 10%/90% to 90%/10%. The results are shown in Fig. 7. They show that in situations with a high fraction of best-effort traffic preemption leads to a substantial decrease of the blocking probabilities for the QoS users with still acceptable dropping probabilities. However, if the ratio is shifted to the QoS side, the decreasing load available to the E-DCH users leads to increased dropping probabilities.

Fig. 8 shows the sensitivity of the system to different volume size distributions for the E-DCH users. The results are calculated with an event-based simulation which was also used for the validation of the analytical results. Three cases are presented: Constant volume size, exponentially and Pareto distributed volume sizes (with parameters $k = 1.5$ and $x_m = 2.4 \cdot 10^4$), all with the same mean. As expected (see e.g. [13]), a higher variance leads to lower dropping probabilities, although in this very low load regions the differences are quite small, which may lead to the conclusion that the exponential assumption may be a sufficient approximation in these cases. Of course, this should be carefully validated.

8 Conclusion

We presented an analytical model for QoS and best-effort traffic over the enhanced uplink under the assumption of two admission control policies, *preserving* and *preemptive*. The model includes the effects of imperfect power control and lognormal distributed other-cell interference. The model of the admission control uses a load-based approach, i.e. connects the primarily limiting shared resource, which is the multiple access interference in the uplink, to the blocking and dropping probabilities. The evaluation of the two admission control policies showed that preemption can lead to a substantial decrease in blocking probabilities for the QoS users, but it should be generally carefully used since it can also lead to high dropping probabilities in scenarios with low quantities of best-effort

traffic. A possible solution to this would be to reserve a certain amount of load to best-effort users only.

Acknowledgments:

The authors thank Prof. Phuoc Tran-Gia and Tobias Hoßfeld, University of Würzburg, Dr. Hans Barth, T-Mobile International, and Tuo Liu, University of Sidney for the fruitful discussions.

References

1. 3GPP: 3GPP TS 25.309 V6.4.0 FDD enhanced uplink; Overall description; Stage 2. Technical report, 3GPP (2005)
2. Parkvall, S., Peisa, J., Torsner, J., Sågfors, M., Malm, P.: WCDMA Enhanced Uplink – Principles and Basic Operation. In: Proc. of VTC Spring '05, Stockholm, Sweden (2005)
3. Mäder, A., Staehle, D.: An Analytical Model for Best-Effort Traffic over the UMTS Enhanced Uplink. In: Proc. of IEEE VTC Fall '06. (2006)
4. Altman, E.: Capacity of Multi-Service Cellular Networks with Transmission-Rate Control: A Queueing Analysis. In: Proc. of MobiCom '02, Atlanta, Georgia, USA (2002) 205–214
5. Fodor, G., Telek, M.: Performance Analysis of the Uplink of a CDMA Cell Supporting Elastic Services. In: Proc of NETWORKING 2005, Waterloo, Canada (2005) 205–216
6. Litjens, R., Boucherie, R.J.: Performance Analysis of Fair Channel Sharing Policies in an Integrated Cellular Voice/Data Network. Telecommunication Systems **19**(2) (2002) 147–186
7. Holma, H., Toskala, A.: WCDMA for UMTS. John Wiley & Sons, Ltd. (2001)
8. Viterbi, A., Viterbi, A.: Erlang Capacity of a Power Controlled CDMA System. IEEE Journal on Selected Areas in Communications **11**(6) (1993) 892–900
9. Fenton, L.F.: The sum of log-normal probability distributions in scatter transmission systems. IRE Transactions on Communication Systems **8**(1) (1960) 57–67
10. Staehle, D., Leibnitz, K., Heck, K., Tran-Gia, P., Schröder, B., Weller, A.: Analytic Approximation of the Effective Bandwidth for Best-Effort Services in UMTS Networks. In: Proc. of VTC Spring '03, Jeju, South Korea (2003)
11. 3GPP: 3gpp ts 25.321 v6.6.0 medium access control (mac) protocol specification. Technical report, 3GPP (2005)
12. Hossfeld, T., Mäder, A., Staehle, D.: When do we need Rate Control for Dedicated Channels in UMTS? In: Proc. of VTC Spring '06. (2006)
13. Litjens, R., Boucherie, R.J.: Elastic calls in an integrated services network: the greater the call size variability the better the QoS. Performance Evaluation **52**(4) (2003) 193–220

A New Model for Public-Key Authentication

Reto Kohlas, Jacek Jonczy, and Rolf Haenni

Institute of Computer Science and Applied Mathematics
University of Berne
3012 Berne, Switzerland
{kohlas,jonczy,haenni}@iam.unibe.ch

Abstract PGP's Web of Trust, Maurer's model of a PKI, Jøsang's Certification Algebra, Haenni's key validation method, and Credential Networks provide techniques for authenticating a public key in a decentralized public-key infrastructure. They allow to compute a degree of confidence for the authenticity of someone else's public key. The computations are generally based on a reasoner's trust assumptions and evidence such as public-key certificates and recommendations. In this paper, a new model and a formal language applicable to decentralized public-key authentication are introduced. The model is intended to serve as a basis for more sophisticated public-key authentication methods, as it incorporates some important points neglected so far. For example, the possibility of a physical entity to use different keys, to act under different names, or to share a key with others is considered.[1]

1 Introduction

Public-key cryptography [3,13] allows to encrypt and sign messages. It is deployed in a wide range of today's network and e-commerce applications. One of its advantages is that each entity has to generate and protect only *one* key-pair. But although public keys can be distributed in a non-secret manner, it must be guaranteed that they reach other entities in an *authentic* way. This is often difficult to achieve and referred here to as the *public-key authentication* problem.

One possibility to solve this problem is that so-called *certification authorities (CA's)* or *introducers* issue *public-key certificates*. These are digitally signed statements attesting the public key's authenticity for a physical entity. Certificates are distributed either in a *centralized* or *decentralized* public-key infrastructure. In a hierarchical, centralized system like X.509, the entities that act as CAs are often designated by law or by an organization's policy. In such a context, a CA is normally trusted by default. In the decentralized approach, on the other hand, all entities can issue certificates. A user of such a system faces the delicate question whom she should trust, and for which type of statements. A prominent example of a decentralized public-key authentication system is PGP's Web of Trust [18,14].

[1] This research is supported by the Hasler Foundation, project no. 2042, and the Swiss National Science Foundation, project no. PP002–102652.

1.1 Decentralized Public-key Authentication

Decentralized public-key authentication generally depends on several *pieces of evidence* and *assumptions*. The evidence comes in form of public-key certificates, recommendations, and discredits. The assumptions concern the trustworthiness (honesty and competence) of the introducers, and the authenticity of public keys directly obtained from other entities.

Evidence and assumptions for public-key authentication can be *uncertain*. For instance, one perhaps *not completely* trusts or distrusts some physical entity[2], but only to some degree. Many methods therefore allow a physical entity to assign a *confidence value* to each of her pieces of evidence and assumptions. Such a confidence value is meant to stand for her degree of belief with respect to the judged assertion and is often an element of an ordered, discrete set (e.g., {`no trust, marginal trust, full trust`}), or then a real number between 0 and 1.

There are at least the following two tasks necessary for devising a public-key authentication method (Maurer for instance distinguishes between the deterministic and the probabilistic part of his model [10]):

- **Defining the model.** It must be made precise which evidence is considered to be explicit part of the model. For instance, does the model capture the possibility that a physical entity might be using different keys or that one public-key entity is used by different physical entities simultaneously? Should recommendations and discredits be integrated into the model? Does the model deal with negative evidence? For instance, the statement that a public-key entity is not authentic for a physical entity can constitute valuable information in some situations.
- **Handling confidence values.** The question is how confidence values representing a physical entity's degree of belief with respect to the pieces of evidence should be reasonably combined. Whereas some methods rely on well-founded frameworks such as probability theory or probabilistic argumentation, other methods implement their own, ad hoc strategies.

As explained in Subsection 1.3, the scope of this paper is finding an appropriate model for public-key authentication,

1.2 Motivation

PGP's web of trust has obviously some conceptual deficiencies and weaknesses [10,9,15]. As a consequence, different public-key authentication methods have been proposed in the last decade. Examples are, amongst others, Maurer's probabilistic model of a public-key infrastructure [10], Haenni's key validation method [5], Jøsang's Certification Algebra [7], and Credential Networks [6].

These methods do not only combine confidence values in different ways, but they also make use of different *models*. For instance, Credential Networks allow a user to state explicitly that another entity should not be trusted.[3] Other methods

[2] The entity is called *physical* entity for reasons that are explained later.

[3] Which corresponds to a negative recommendation, and is called *discredit* in [6].

do not yet consider such negative evidence. As the following examples show, there are a number of different modeling issues that need to be investigated further:

- **Public-key authenticity.** Public-key authenticity is most of the times seen as a one-to-one relationship between physical and public key entity [10]. The realistic possibility that one physical entity uses several keys is neglected. However, detecting such possibly hidden dependencies in key ownership is important, since it allows to conclude that two statements signed by different keys actually stem from the same entity. Similarly, the case in which two different entities claim or prove ownership for the *same* key has not yet received attention, to our knowledge.
- **Trust and distrust.** In the methods proposed so far, trust is solely considered to be something "positive". There it is a mechanism by which a statement, made by the trusted entity, is validated. However, it is possible, sometimes even probable, that malicious entities try to willfully spread false information. Under the assumption that another entity is malicious it can make sense to conclude the converse of what the suspect says. This kind of distrust has not yet been taken into account.
- **Descriptions and identifiers.** A public-key certificate is often defined to be a digitally signed statement that binds a *public key* to an *entity* (for instance, see [10]). However, assertions can only be made about *names* or *descriptions* of these principals. The case that some physical entity is referred to by different identifiers in a certain context must be analyzed.

The fact that some aspects are left out by the existing models is the motivation to search for a more general model of public-key authentication.

1.3 Goals and Outline

To our knowledge, most methods introduce a set of basic units of concern, called statements or propositions, and give them an informal meaning. In this paper, the problem of finding a model for a public-key authentication method is tackled in a different way.

We express the main concepts such as physical, digital, and public-key entities, and their attributes and relationships in terms of an *entity-relationship (ER) model* (Section 2). The ER-modeling leads to a *formal language* in Section 3. The language is flexible in the sense that it allows to formalize different semantics for concepts such as public-key authenticity and trust. Moreover, it allows to model aspects neglected so far. This is illustrated in Section 4. The paper concludes in Section 5 by mentioning directions for future research.

The here presented language appears - at first sight - to be similar to logics such as BAN [1] and GNY [4], but the scope of our work is different. Whereas BAN and GNY are analysis tools for cryptographic protocols, we intend to devise a logic allowing to formalize concepts for decentralized public-key authentication. Our model can be seen as continuation and refinement of previous models, in particular the methods mentioned in Subsection 1.2. Also note that the aim of

the paper is not to devise directly a new public-key authentication method; this is future work, and it will be based on the here presented model.

2 An ER-Model for Public-Key Authentication

ER-models [2] have been successfully used in many different areas of Computer Science. Applying them to modeling public-key authentication is therefore a straightforward idea.

We have attempted to incorporate different aspects relevant for public-key authentication in the model. It is intended to contain only the *basic* concepts; the idea is that other, more abstract concepts can be derived from the basic one's. As an example, the semantics of public-key authenticity for a physical entity can depend on the application. The definition for public-key authenticity can be derived from control of public-key entity, which is a basic relationship of the model.

2.1 Principal Sets and Character Strings

The *principals* or *entities* of our model are either physical, digital, or public-key entities (see Figure 1). By *physical world entity* we mean an entity that exists in the "reality of the physical world". For simplicity, physical world entities are referred to as *physical entities* in this paper. Examples of physical entities are persons, agents, systems, groups, organizations, and companies. They can control public-key and digital entities (see Subsection 2.3). Physical entities may have some computational power used for the generation and verification of digital signatures. The set of entities is denoted by $\mathcal{E}$.

By *digital entity* we refer to a communication unit in a networked environment like the Internet. Although digital entities may appear as being autonomous, they are in fact always controlled by some physical entity. Examples of digital entities are web and ftp sites, e-mail accounts, or participants of e-commerce systems like eBay. The set of digital entities is denoted by $\mathcal{V}$.

A *public-key entity* is a principal used for digital signature generation and verification.[4] It consists of a public and a private key. Access to the private key allows to generate signatures in the public-key entity's name. Therefore unauthorized access to the private key has to be prevented. Knowledge of the public key allows the verification of a digital signature. The set of public-key entities is denoted by $\mathcal{K}$.

Finally, let $\mathcal{A}$ stand for a set of character strings, and let $\mathbb{N}$ as usually denote the set of the natural numbers.

[4] Encryption public-key entities are not explicitly included in the model, for simplicity reasons. They could be integrated into the model without big effort.

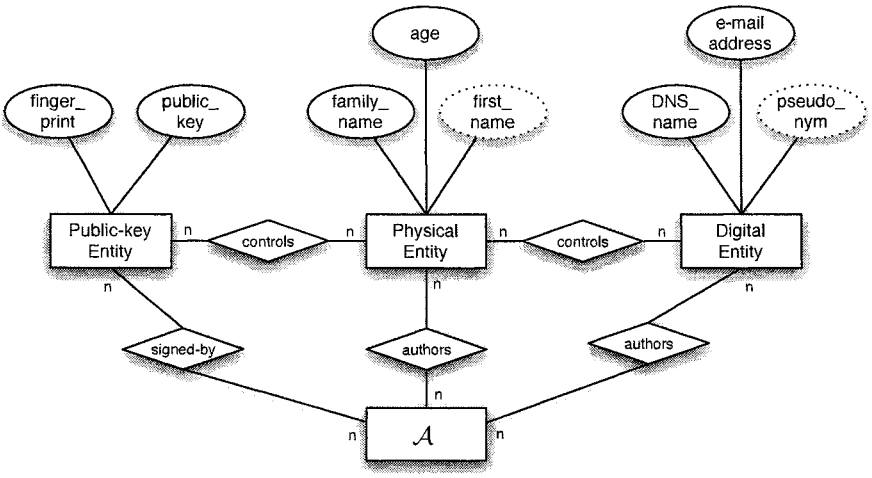


Figure 1. An ER-model for public-key authentication.

2.2 Attributes and Naming

Principals are characterized by attributes, which allow to name and to make assertions about them. For illustration, we introduce a number of attributes. Note that for a specific application it can naturally make sense to model additional or different attributes.

For example, we introduce a physical entity's attribute *family_name*. The attribute can be represented as subset of the Cartesian Product between physical entities and character strings: $family_name \subseteq \mathcal{E} \times \mathcal{A}$. We define its semantics as follows: "$(E, A) \in family_name$, where $E \in \mathcal{E}, A \in \mathcal{A}$, if and only if A is the family name of E as specified in E's passport (or any other official document)."[5]

An example of an attribute for a public-key entity is $fingerprint$:"$(K, A) \in fingerprint$, where $K \in \mathcal{K}, A \in \mathcal{A}$, if and only if A is the string that is obtained by applying the SHA hash function as described in [12] to the public key of K." A possible attribute for a digital entity is its DNS-name (for a server), a pseudonym (for an e-commerce participant) or an email address (for an email account).

Multi-valued attributes are attributes that can take several values at the same time. An example is the attribute $first_name$, since a person can have different first names.

[5] If E is not a natural person, she has no passport, and in this case trivially $(E, A) \notin family_name$.

2.3 Relationships

Relationships capture relations between the entities and character strings. A physical entity *controls* a public-key entity whenever she has access to its private key. In general, access to a private key occurs through knowledge of a password or passphrase, or through possession of a physical device like a smartcard. It is possible that the same key is controlled by more than one physical entity, or that a physical entity controls more than one key. The relationship is denoted by $controls_k$: $controls_k \subseteq \mathcal{E} \times \mathcal{K}$.

A physical entity may not only control public-key, but also digital entities. Control is therefore also a relationship between a physical and a digital entity: $controls_d \subseteq \mathcal{E} \times \mathcal{V}$. It means that the physical entity can determine or at least influence the digital entity's behavior. Control again happens generally through knowledge of a password. On the eBay web site, for instance, by providing her username and password, a physical entity can rate an Ebay entity in the name of the digital entity she controls.

That a digital signature for a statement has been generated by a public-key entity is represented by the relationship *signed-by* (or *signs*). Note that this relationship does not capture which physical entity was using the key. It is a binary relationship between a public-key entity and a statement: $signs \subseteq \mathcal{K} \times \mathcal{A}$.

Authors is a relationship between a physical entity and a statement. Here, *authors* is not only meant in its literal meaning. It rather stands for the fact that it was in a physical entity's intention to be considered as the source of the message. In our model, authoring a statement can mean pronouncing it, writing it on a piece of paper, typing it into a computer, or creating any other representation of it. One purpose of the digital signature is to provide evidence that a physical entity was *authoring* the signed statement. The relationship is denoted by $authors_e$, where $authors_e \subseteq \mathcal{E} \times \mathcal{A}$. In the sequel, we will sometimes use *says* equivalently for authors.

A digital entity *authors* a statement if the statement actually stems from the digital entity. A web server for instance *authors* a statement if that statement is part of its web site content. As another example, an email account *authors* if the message really stems from the email account's holder. The relationship is denoted by $authors_d \subseteq \mathcal{V} \times \mathcal{A}$.

3 An Evidential Language $\mathcal{S}$

We introduce a language $\mathcal{S}$ for representing a physical entity's observations, assumptions, pieces of evidence and conclusions in the context of public-key authentication. $\mathcal{S}$ allows to express different meanings for public-key certificates, public-key authenticity, and trust. Moreover, using $\mathcal{S}$ it is possible to describe the logical relationship between these notions.

A statement $s \in \mathcal{S}$ can for instance stand for the following: "There is someone saying that a physical entity E whose family name is *Brown* and whose first name is *John* claimed to have control of a public-key entity K, whose fingerprint is *3AC4LPQA*".

The language $\mathcal{S}$ is an instance of a many-sorted first order logic (MSFOL). Introducing MSFOL is relatively complex, since one has to formalize many different semantical concepts such as signatures, interpretations, free and bound occurrences of variables, and valuations. Due to space limitations, we only illustrate the basic ideas behind $\mathcal{S}$ by discussing some examples and and refer to [11,17] for a detailed and formal account of MSFOL, and to [16] for a concrete MSFOL-application.

An example of a expression that contains many linguistic elements of $\mathcal{S}$ is

$$\exists E \exists K(controls_k(E, K) \wedge fingerprint(K, A...1Z) \wedge first_name(E, \text{“}John\text{”})).$$

Both attributes and relationships of the ER-model are in $\mathcal{S}$ represented by predicates. For example, $controls_k$ is a predicate of the relationship $controls_k$, and $fingerprint$ is a predicate for the public-key entity attribute $fingerprint$. The arguments of the predicates are variables of the corresponding sort. E is a variable standing for a physical entity, and K stands for a public-key entity. If the argument of a predicate is of sort $\mathcal{A}$ or $\mathbb{N}$, it can be a constant term. For instance, *"A3LMZHH1Z"* is a constant term standing for the corresponding character string. The quantifiers $\exists$ and $\forall$, and the logical connectives such as $\wedge$, $\vee$, $\neg$, $\rightarrow$ and $\leftrightarrow$ are used as common. The meaning of the above statement is therefore: There exists a physical entity E ($\exists E$) and a public-key entity K ($\exists K$) such that E controls K and ($\wedge$) the fingerprint of K is *"A...1Z"*, and the first name of E is "John".

It is common in MSFOL to attach a sort to the quantifiers. For instance, we could have written $\exists_{keypair} K$ instead of $\exists K$ to indicate that K is of sort *keypair*. Because in $\mathcal{S}$ it is always clear by the variable name of which sort the variable is, we have chosen to use a simpler notation omitting the subscripts.

We will use the equality symbol "$=$" to denote that two principals are identical. For instance, we could state that a public key K, whose fingerprint corresponds to "$A \ldots 1Z$", is controlled by two *different* entities E_1 and E_2 (and hence $\neg(E_1 = E_2)$):

$$\exists E_1, E_2, K(controls_k(E_1, K) \wedge controls_k(E_2, K) \\ \wedge \neg(E_1 = E_2) \wedge fingerprint(K, \text{“}A...1Z\text{”})).$$

In the examples that follow, we will sometimes use free occurrences of variables for $\mathcal{A}$. For instance, the first example will be written as

$$\exists E \exists K(controls(E, K) \wedge fingerprint(K, A) \wedge first_name(E, \text{“}John\text{”})),$$

where A stands for a particular character string, which will not be concretely indicated.

4 Entity Descriptions and Evidence for Public-Key Authentication

In this Section we illustrate the key role of descriptions and identifiers in public-key authentication. Moreover, we provide possible semantics for public-key authenticity and trust.

4.1 Entity and Description Space

The goal of public-key authentication is to validate the authenticity of a *public key* for a *physical entity*. But public-key certificates and assertions are always with respect to *descriptions, identifiers* or *names* of these principals. One is therefore in principle confronted with two different reasoning levels. The first one is the *universe of entities* consisting of the physical entities, digital entities, and public-key entities as they exist in the "real world" or in our minds. The other is the *universe of descriptions* that is made up of the descriptions and identifiers used for the principals. In the universe of descriptions, the principals are represented just by some pieces of information which are provided by descriptions or identifiers.

In this paper, an entity description, or description for short, is a logical formula α in $\mathcal{S}$ that characterizes an entity. The formula expresses that the principal has some concrete values for its attributes or is involved in a certain relationship. Technically, α is a formula that has exactly one *free* variable.[6] The following example shows a description for a physical entity, consisting both of attributes and relationships:

$$family_name(E, \text{"Smith"}) \wedge first_name(E, \text{"Bob"}) \\ \wedge controls(E, \text{"bsmith@gm.com"}).$$

The above description has one free variable, which is E. The description stands for a physical entity "Bob Smith", who controls the mail account "bsmith@gm.com". In the sequel, let α_X stand for a description whose free variable is X.

Existing methods do not differentiate between universe of entities and descriptions. In Maurer's model [10], for example, $Aut_{A,B}$ "denotes the statement that, from A(lice)'s point of view, her copy of B's public key is authentic."

4.2 Identifiers

A public-key authentication method actually provides decision support only for the binding between the two descriptions of a physical and a public-key entity, rather than directly for entities. Likewise, a public-key certificate can strictly speaking only attest a relationship between descriptions of these entities. But someone that uses public-key cryptography wants sometimes to securely communicate with concrete physical entities that are in his mind. The goal is then to ascribe a public-key entity or a statement to a physical entity, not solely to its description. It must therefore be guaranteed that in a certain context a description matches only *one* single entity. Such a unique description is called an *identifier* in this paper.

Finding a naming scheme that provides identifiers for all entities involved in a certain system is often difficult. In particular, finding globally known identifiers for natural persons can be nontrivial. The problem is to choose a set of attributes that are easy verifiable, meaningful, and respect the principles of data protection.

[6] A variable X is free in s, if it has at least one free occurrence in s. An occurrence of X is free in s, if it is not bound by any quantifier.

The fact that a description α_X is an identifier constitutes a hypothesis, which can be expressed in the language $\mathcal{S}$ as follows (α_X is to be substituted by the concrete description):

$$\forall X_1 \forall X_2 \big(((\alpha_{X_1} \wedge \alpha_{X_2}) \rightarrow (X_1 = X_2)) \wedge (\exists X \alpha_X) \big).$$

The implication in the quantified formula states that if there are two entities X_1 and X_2 which satisfy their respective descriptions, then they must actually be identical ($X_1 = X_2$). This implies that the description α_X stands for at most one entity. The rightmost subformula requires the existence of at least one entity that matches the description. This finally means that there is exactly one entity that matches the description, and the description α_X is therefore an identifier.

In PGP's Web of Trust, a physical entity is often described by its first and last name, and by an e-mail address. It makes sense to include the e-mail address, since this increases the likeliness of the so-obtained description to be unique. One problem, however, is that in general it is not verified by the introducers whether the certified entity really controls the email address. In this sense it is questionable whether such a description really matches the certified entity.

4.3 Equality between Identifiers

Assigning trust values to public keys, in order to rate the trustworthiness of an introducer, is problematic [9]. The reason is that in such a way, a physical entity can act in the name of multiple entities, just by using different signature public keys. Maurer therefore suggests to assign trust values to physical entities[7]. In this way, dependencies in key ownership are considered.

As in the universe of descriptions entities appear in the form of identifiers, one has to assign trust values to identifiers. Again, a dependency problem arises. Entities may be referred to by different identifiers in a closed system (e.g., web of trust). It is crucial to detect such dependencies and to figure out which identifiers actually refer to the same entity.

The equivalence of two descriptions is therefore another important hypothesis, which is also expressible in $\mathcal{S}$. Let α_X^1 and α_X^2 stand for two descriptions representing two entities of the same sort. Two descriptions are *semantically equivalent*, or *equivalent* for short, if they stand for the same entity:

$$\forall X (\alpha_X^1 \leftrightarrow \alpha_X^2).$$

That is, α_X^1 and α_X^2 are equivalent if and only if an entity satisfies α_X^2 when it satisfies α_X^1, and conversely.

The question is what evidence is available allowing conclusions about the equivalence of identifiers. It is conceivable to extend an introducer's role in the context of public-key authentication. For instance, an introducer could issue an alternative kind of certificates stating that two identifiers are actually equivalent.

[7] In Maurer's paper, the notion of *entity* is close to what is called *physical entity* in our paper.

4.4 Public-key Authenticity

In the following, we put the problems of determining identifiers and checking the equivalence between identifiers aside. We assume that all entities have exactly *one sole* and commonly known identifier. Because of this one-to-one correspondence between entities and identifiers, we can represent the formulas directly in terms of the entities.

A public-key entity K is authentic for a physical entity E if K "speaks for" for E, or in other words, if the successful verification of a digital signature by K under a message, allows the conclusion that it was in E's intention to be considered as the source of the signed messages. This is captured by the following formula:

$$aut(E, K) \rightarrow (signs(K, S) \rightarrow authors(E, S)).$$

Note that this definition is equivalent to

$$(aut(E, K) \wedge signs(K, S)) \rightarrow authors(E, S).$$

By introducing authenticity in this way, it is taken into account that one physical entity uses different public-key entities.

But which concrete evidence must be collected in order to accept a key pair K to be authentic for E? If one meets E in person, E could indicate the public key of K. In this case, the following piece of evidences would be available, represented by the following statement:

$$controls_k(E, K) \rightarrow aut(E, K).$$

The question is whether one really needs to verify that E is the (sole) owner of K. Nobody can prevent E from giving away control over K to another entity, and one will anyway not be able to ascertain that E is the sole owner of K. Fortunately, it is not in E's interest to claim control of a public key she does not really control (otherwise the real key owner could make statements in E's name)[8]. A reasonable meaning for public-key authenticity therefore looks as follows: if E says that she controls a public-key entity K whose fingerprint is A, then the public key can be accepted as being authentic for E. This is captured by the following statement:

$$authors(E, controls_k(E, K)) \rightarrow controls_k(E, K).$$

Another case occurs when different physical entities claim ownership for the same public-key pair. Such a situation can be very confusing, since a digitally signed message cannot be assigned unequivocally to a certain physical entity anymore. One possible solution is that such public-key sharing is prohibited by the system's policy. This is expressed by the following statement:

$$\neg(E_1 = E_2) \rightarrow \neg(aut(E_1, K) \wedge aut(E_2, K)).$$

The formula expresses that K can not be authentic for two different physical entities E_1 and E_2 simultaneously.

4.5 Trust and Distrust

So far, trust has been defined as being an autonomous statement, whose semantics has been informally described. However, as the following formula illustrates, a possible formal trust semantics is obtained by expressing it based on notions defined within our model:

$$trust(E, S) \rightarrow (authors(E, S) \rightarrow S).$$

Here, the variable S stands for a word of the language S. By this definition, trust in E with respect to the statement S allows to conclude the validity of S in case E authors S. This allows to express trust with respect to different types of statement. For instance, an introducer E_1 may try to warn other participants of the system that some public-key entity is *not* authentic for an entity E_2:

$$authors(E_1, \neg aut(E_2, K)).$$

Of course, E_1 could also make the statement $\neg aut(E_2, K)$ by digitally signing it using her public-key entity.

Entities who distribute such types of warnings have to be trusted for this kind of statements, which is expressed as follows:

$$trust(E_1, authors(E_1, \neg aut(E_2, K))).$$

As mentioned in Subsection 1.2, it is justifiable to explicitly disbelieve in what a physical entity says. According to the above formula, if an entity is distrusted ($\neg trust(E, S)$), no conclusion is possible, even if $authors(E, S)$ is valid. An alternative trust definition could be the following:

$$trust(E, S) \leftrightarrow (authors(E, S) \rightarrow S)$$

In this case, if $\neg trust(E, S)$ and $authors(E, S)$ are both valid, one is able to conclude the negation of S, namely $\neg S$.

5 Conclusion

We have introduced an ER-model and a language $\mathcal{S}$ making it possible to represent evidence and conclusions in the context of public-key authentication. The language has allowed us to point out different subtleties in notions such as public-key authenticity and trust. One main aspect discussed in this paper is that public-key authentication methods in fact always rely only on information about entities, not the entities themselves. A description provides such information and can be expressed as a formula of our language. A special kind of a description standing for one sole entity is an identifier. We have argued that deciding whether a description is an identifier and whether two identifiers represent the same entity are two key tasks for decentralized public-key authentication. The suggestions for improvement identified here will allow to revise existing methods as part of future work, and make it possible to propose an enhanced public-key authentication method.

References

1. Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990.
2. Peter Pin-Shan S. Chen. The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.
3. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.
4. Li Gong, Roger Needham, and Raphael Yahalom. Reasoning About Belief in Cryptographic Protocols. In Deborah Cooper and Teresa Lunt, editors, *Proceedings 1990 IEEE Symposium on Research in Security and Privacy*, pages 234–248. IEEE Computer Society, 1990.
5. R. Haenni. Using probabilistic argumentation for key validation in public-key cryptography. *International Journal of Approximate Reasoning*, 38(3):355–376, 2005.
6. J. Jonczy and R. Haenni. Credential networks: a general model for distributed trust and authenticity management. In A. Ghorbani and S. Marsh, editors, *PST'05: 3rd Annual Conference on Privacy, Security and Trust*, pages 101–112, St. Andrews, Canada, 2005.
7. A. Jøsang. An algebra for assessing trust in certification chains. In *NDSS'99: 6th Annual Symposium on Network and Distributed System Security*, San Diego, USA, 1999.
8. R. Kohlas and U. Maurer. Reasoning about public-key certification: On bindings between entities and public keys. In M. K. Franklin, editor, *Financial Cryptography'99, Third International Conference*, LNCS 1648, pages 86–103, Anguilla, British West Indies, 1999. Springer.
9. R. Kohlas and U. Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In H. Imai and Y. Zheng, editors, *PKC'2000, Third International Workshop on Practice and Theory in Public Key Cryptography*, LNCS 1751, pages 93–112, Melbourne, Australia, 2000. Springer.
10. U. Maurer. Modelling a public-key infrastructure. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, *ESORICS, European Symposium on Research in Computer Security*, LNCS 1146, pages 324–350. Springer, 1996.
11. K. Meinke and J. V. Tucker, editors. *Many-sorted logic and its applications.* John Wiley & Sons, Inc., New York, NY, USA, 1993.
12. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, USA, 1996.
13. R. L. Rivest, A. Shamir, and L. M. Adelman. A method for obtaining digital signatures and public-key cryptosystems. Technical Report TM-82, MIT, Cambridge, USA, 1977.
14. W. Stallings. *Protect Your Privacy, a Guide for PGP Users.* Prentice Hall, 1995.
15. M. Wüthrich. GnuPG and probabilistic key validation. Bachelor thesis, IAM, University of Berne, Switzerland, 2006.
16. J. S. H. Yang, Y. H. Chin, and C. G. Chung. Many-sorted first-order logic database language. *Comput. J.*, 35(2):129–137, 1992.
17. Calogero G. Zarba. Many-sorted logic. http://theory.stanford.edu/~zarba/snow/-ch01.pdf.
18. P. R. Zimmermann. *The Official PGP User's Guide.* MIT Press, 1994.

A Proof of Concept Implementation of SSL/TLS Session-Aware User Authentication (TLS-SA)

Rolf Oppliger[1], Ralf Hauser[2], David Basin[3], Aldo Rodenhaeuser[4] und Bruno Kaiser[4]

[1] eSECURITY Technologies, Beethovenstrasse 10, CH-3073 Gümligen
[2] PrivaSphere AG, Jupiterstrasse 49, CH-8032 Zürich
[3] Department of Computer Science, ETH Zurich, Haldeneggsteig 4, CH-8092 Zürich
[4] AdNovum Informatik AG, Röntgenstrasse 22, CH-8005 Zürich

Abstract Most SSL/TLS-based e-commerce applications employ conventional mechanisms for user authentication. These mechanisms—if decoupled from SSL/TLS session establishment—are vulnerable to man-in-the-middle (MITM) attacks. In this paper, we elaborate on the feasibility of MITM attacks, survey countermeasures, introduce the notion of SSL/TLS session-aware user authentication (TLS-SA), and present a proof of concept implementation of TLS-SA. We think that TLS-SA fills a gap between the use of public key certificates on the client side and currently deployed user authentication mechanisms. Most importantly, it allows for the continued use of legacy two-factor authentication devices while still providing high levels of protection against MITM attacks.

1 Introduction

Most electronic commerce (e-commerce) applications in use today employ the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol [1] to authenticate the server to the client and to cryptographically protect the communication channel between them. When setting up a secure channel, the main point where SSL/TLS-based applications actually differ concerns user authentication. Conventional options available include passwords, personal identification numbers (PINs), transaction authorization numbers (TANs), scratch lists, as well as more sophisticated authentication systems, such as one-time password (OTP) and challenge-response (C/R) systems. In contrast, there are only a few applications that employ client-side public key certificates for user authentication as part of the SSL/TLS session establishment. In fact, the deployment of public key certificates has turned out to be slow—certainly slower than it was originally anticipated [2]. This is particularly true for client-side authentication.

In spite of the fact that many researchers have analyzed the security of the SSL/TLS protocol (e.g., [3, 4]), relatively few shortcomings and vulnerabilities have been identified (e.g., [5–7]. Most of them are theoretically interesting but not practically relevant. More problematic is the possibility that the protocol may be used in some inappropriate way or that it may be used in an environment that makes its security properties meaningless because the actual threats are

different than the ones assumed in the design phase (e.g., [8]). This paper further illustrates this point.

The vast majority of SSL/TLS-based e-commerce applications employ conventional user authentication mechanisms on the client side, and hence are vulnerable to phishing, Web spoofing, and—most importantly—man-in-the-middle (MITM) attacks [9].[1] These attacks are particularly powerful if visual spoofing is employed [10]. If a MITM can place himself between the user and the server, then he can act as a relay and authenticate himself to the server on behalf of the user. Even worse, if the MITM operates in real-time, then there is hardly any user authentication mechanism (decoupled from SSL/TLS session establishment) that cannot be defeated or misused. This fact is often neglected when considering the security of SSL/TLS-based e-commerce applications, such as Internet banking or remote Internet voting.

In this paper, we elaborate on the feasibility of MITM attacks in Section 2, survey countermeasures and related work in Section 3, introduce the notion of SSL/TLS session-aware user authentication (TLS-SA) in Section 4, overview a proof of concept implementation of TLS-SA in Section 5, and discuss some weaknesses and limitations thereof in Section 6. Finally, we finish the paper with conclusions and an outlook in Section 7. In summary, we think that TLS-SA fills a gap between the use of public key certificates on the client side and currently deployed user authentication mechanisms. Most importantly, and in contrast to related work, it allows for the continued use of legacy two-factor authentication devices while still providing high levels of protection against MITM attacks.

2 MITM Attacks

According to RFC 2828, a *MITM* attack refers to "a form of active wiretapping attack in which the attacker intercepts and selectively modifies communicated data in order to masquerade as one or more of the entities involved in a communication association." Consequently, the major characteristics of MITM attacks are that they represent active attacks, and that they target the associations between the communicating entities (rather than the entities or the communication channels). Note that in the literature, a MITM that carries out an active attack in real-time is sometimes also called *adaptive*. We will not use this term and MITM attacks will be adaptive by default.

In an SSL/TLS setting, which is standard in Internet banking and other Web-based e-commerce applications, the best way to think about a MITM attack is to consider an adversary that represents an SSL/TLS proxy server (or relay) between the client and the server. Neither the client nor the server are aware of the MITM. Cryptography makes no difference here as the MITM is in the loop and can decrypt and re-encrypt on the fly all messages that are sent back and

[1] According to http://www.theregister.co.uk/2005/10/12/outlaw_phishing, a MITM attack was launched against a Swedish Internet bank in November 2005. More recently, a MITM attack was launched against the Internet banking users of Citibank (as reported on July 10, 2006, by Brian Krebs in his Washington Post blog).

forth. Also, a MITM need not operate alone. In fact, there may be many entities involved in a MITM attack. One entity may be located between the client and the server, whereas some other entities may be located elsewhere. In this case, the corresponding attacks are called *Mafia* or *terrorist fraud* [11], and the underlying problem is referred to as the *chess grandmaster problem*. In this paper, we do not care whether the adversary is acting as a single-entity or multiple-entity MITM. Instead, we use the term MITM attack to refer to all types of attacks in which at least one entity is logically located between the client and the server.

In our SSL/TLS setting, there are many possibilities to implement a MITM attack. The user may be directed to the MITM using standard phishing techniques. Other possibilities include Address Resolution Protocol (ARP) cache poisoning, or Domain Name System (DNS) spoofing (including, for example, pharming). Anyway, MITM attacks may be devastating. If, for example, the user authenticates himself to an application server, then he reveals his credentials to the MITM and the MITM can misuse them to spoof the user. If, for example, the user employs an OTP system, then the MITM can grab the OTP (which is typically valid for at least a few seconds) and reuse it to spoof the user. If the user employs a C/R system, then again the MITM can simply send back and forth the challenge and response messages. Even if the user employed a zero-knowledge authentication protocol [12], then the MITM would still be able to forward the messages and spoof the user. The zero-knowledge property does not, by itself, provide protection against MITM attacks—it only protects against information leakage related to the user's secret.[2]

Given the above, it is perhaps not surprising that most currently deployed user authentication mechanisms fail to provide protection against MITM attacks, even when they run on top of the SSL/TLS protocol. We see two main reasons for this failure:

1. SSL/TLS server authentication is usually done poorly by the naïve end user, if done at all.
2. SSL/TLS session establishment is usually decoupled from user authentication.

The first reason leads to a situation in which the user talks to the MITM, thereby revealing his credentials to the MITM. The second reason means that the credentials can be reused by the MITM to spoof the user to the origin server. We conclude that any effective countermeasure against MITM attacks in an SSL/TLS setting must address these problems by either enforcing proper server authentication or combining user authentication with SSL/TLS session establishment. The first possibility requires hard-coded server certificates or dedicated client software, whereas the second possibility requires modifications to the SSL/TLS or the authentication protocols in use. For the purpose of this paper, we confine ourselves to the second possibility and focus on modifying the authentication protocol accordingly.

[2] Note, however, that there is a construction [13] that can be used to make a zero-knowledge authentication protocol resistant against MITM attacks.

3 Countermeasures and Related Work

In spite of the fact that MITM attacks pose a serious threat to SSL/TLS-based e-commerce applications, there are only a few countermeasures and surprisingly little related work. Moreover, the situation is often misunderstood: it is often stated within the security industry that strong (possibly two-factor) user authentication is needed to thwart MITM attacks.[3] This statement is flawed. Vulnerability to MITM attacks is not a user authentication problem—it is a server authentication problem. MITM attacks are possible because SSL/TLS server authentication is usually done poorly by the user (see the first point enumerated above). In other words: if users properly authenticated the server with which they establish an SSL/TLS session, then they would be protected against MITM attacks. Unfortunately, this is not the case and it is questionable whether it is possible at all, given the sophistication of typical users. Note that a MITM can employ many tricks to give the user the impression of being connected to an origin server, for example using visual spoofing. In the most extreme case, one may think of a MITM that is able to control the graphical user interface of the browser. Also, we have seen phishing sites that employ valid certificates.[4] In such a setting, most users are out tricked and are unable to recognize that they are subject to a MITM attack.

Along the same line of argumentation, it is also important to note that the SSL/TLS protocol has been designed to natively protect users against MITM attacks. In addition to the requirement that certificate-based server authentication must be done properly, SSL/TLS-based MITM protection requires that all clients have personal public key certificates. This requirement could be relaxed, if one extended the SSL/TLS protocol with some alternative client authentication methods (in addition to public key certificates). For example, there are efforts within the IETF to specify ciphersuites for the TLS protocol that support authentication based on pre-shared secret keys (e.g., [14, 15]) or OTP systems (e.g., [16]). Furthermore, the adoption of password-based key exchange protocols is proposed in [17], and the use of the Secure Remote Password (SRP) is specified in [18]. We think that these efforts are important, but there is a long way to go until the corresponding TLS extensions will be available, implemented, and widely deployed. In the meantime, one cannot count on the security properties of the SSL/TLS protocol alone. Instead, one must assume a setting in which the SSL/TLS protocol is only used to authenticate the server and the user authenticates himself on top of an established SSL/TLS session using conventional authentication mechanisms.

In addition to the SSL/TLS protocol (and extensions thereof), there are several cryptographic techniques and protocols to protect users against MITM attacks:

[3] http://www.antiphishing.org/sponsors_technical_papers/PHISH_WP_0904

[4] We refer to the case in which a phisher employed a valid certificate for www.mountain-america.com and www.mountain-america.net to spoof the Web site of the Mountain America Credit Union at www.mtnamerica.org.

- Rivest and Shamir proposed the Interlock protocol [19] that was later shown to be vulnerable when used for authentication [20].
- Jakobsson and Myers proposed a technique called delayed password disclosure (DPD) that can be used to complement a password-based authentication and key exchange protocol to protect against a special form of MITM attack—called the doppelganger window attack [21]. DPD requires a password-based authentication and key exchange protocol and is not qualified to protect against the kinds of powerful MITM attacks we have in mind.
- Kaliski and Nyström proposed a password protection module (PPM) to protect against MITM attacks [22]. The PPM is a trusted piece of software that utilizes password hashing to generate passcodes that are unique for a specific user and application. Again, PPMs do not provide protection against the kind of MITM attack we have in mind. Furthermore, PPMs appear difficult to deploy and manage.
- Asokan et al. proposed mechanisms to protect tunneled authentication protocols against MITM attacks [23]. These mechanisms are conceptually related to our approach.
- More recently, Parno et al. proposed the use of a trusted device (e.g., a Bluetooth-enabled smartphone) to perform mutual authentication and to thwart MITM attacks [24].

We believe that all of these techniques and protocols either do not adequately protect against MITM attacks or have severe disadvantages when it comes to a large-scale deployment.

Instead of cryptographic techniques and protocols, some researchers have suggested employing multiple communication channels and channel hopping to thwart MITM attacks (e.g., [25]). This approach has its own disadvantages and is impractical for almost all Internet-based applications in use today.

In practice, there is a steadily increasing number of applications—especially in Europe—that authenticate users by sending out SMS messages that contain TANs and require that users enter these TANs when they login. Sending out SMS messages is an example of using two communication channels or two-factor authentication (the mobile phone representing the second factor). While it has been argued that this mechanism protects against MITM attacks, unfortunately, this is not the case. If a MITM is located between the user and the server, then he need not eavesdrop on the SMS messages; all he needs to do to spoof the user is to forward the TAN submitted by the user on the SSL/TLS session. If one wants to work with TANs distributed via SMS messages, then one has to work with transaction-based TANs.[5] For each transaction submitted by the user, a summary must be returned to the user together with a TAN in an SMS message. To confirm the transaction, the user must enter the corresponding TAN. The down-side of this proposal is that transaction-based TANs are expensive (perhaps prohibitively so), they are not particularly user-friendly, and they are not even completely secure (a MITM can still attack the parts of a transaction

[5] http://www.cryptomathic.com/pdf/The Future of Phishing.pdf

that are not part of the summary). We have therefore investigated alternative mechanisms to protect users and their secure sessions against MITM attacks.

4 SSL/TLS Session-Aware User Authentication

Recall from Section 2 that an effective countermeasure against MITM attacks in an SSL/TLS setting must either enforce proper server authentication or combine user authentication with SSL/TLS session establishment. With respect to the first possibility (i.e., enforce proper server authentication), we think that this asks too much of the user. Hence, we confine ourselves to the second approach and examine possibilities for combining user authentication with SSL/TLS session establishment. We use the term *SSL/TLS session-aware user authentication* (TLS-SA) to refer to this approach. The basic idea of TLS-SA is to make the user authentication depend not only on the user's (secret) credentials, but also on state information related to the SSL/TLS session in which the credentials are transferred to the server. The rationale behind this idea is that the server should have the possibility to determine whether the SSL/TLS session in which it receives the credentials is the same as the one the user employed when he sent out the credentials in the first place.

- If the two sessions are the same, then there is probably no MITM involved.
- If the two sessions are different, then something abnormal is taking place. It is then possible or even likely that a MITM is located between the client and the server.

Using TLS-SA, the user authenticates himself by providing a user authentication code (UAC) that depends on both the credentials and the SSL/TLS session (in particular, on information from the SSL/TLS session state that is cryptographically hard to alter). A MITM who gets hold of the UAC can no longer misuse it by simply retransmitting it. The key point is that the UAC is bound to a particular SSL/TLS session. Hence, if the UAC is submitted on a different session, then the server can detect this fact and drop the session.

There are many possibilities to implement TLS-SA, and one can distinguish between hardware-based and software-based implementations.

- In the first case, we are talking about hardware tokens (i.e., hard-tokens). Such a token may be physically connected to the client system or not.[6]
- In the second case, we are talking about software tokens (i.e., soft-tokens).

In either case, an authentication token can be personal or impersonal, and it can be consistent with a cryptographic token interface standard, such as PKCS #11 or Microsoft's CAPI.

[6] We say that the token is physically connected to a system, or just connected, if there is a direct communication path in place between the token and the system. This includes, for example, galvanic connections, as well as connections based on wireless technologies, such as Bluetooth or infrared.

The basic approach to implement TLS-SA was introduced in [26]. It employs impersonal authentication tokens that users can plug into their client systems to support TLS-SA. If such a token is connected, then it is straightforward to provide support for TLS-SA (this is equally true for OTP and C/R systems). In this paper, however, we elaborate on the setting in which the token is not connected to the client system. In this case, the situation is more involved, because the token cannot directly access the hash value of the SSL/TLS session (that is stored in the browser's SSL/TLS cache). Note that there is nothing secret about the hash value (it can, for example, also be retrieved from the network traffic), but it still needs to be accessed in some well-defined way. An obvious possibility is to modify the browser so that it can display the hash value if needed (e.g., in the browser's status bar). Note, however, that the hash value is 36 bytes long, which is far too long to be used entirely (if the users must enter it manually). Consequently, one must reduce the length of the hash value to only a few digits to properly implement TLS-SA in such a setting.

5 Proof of concept Implementation

More recently, we have built a proof of concept implementation of TLS-SA. The aim is to demonstrate the technical feasibility of TLS-SA and to provide a testbed for analysis and further improvement. On the client side, the implementation employs C/R tokens from Vasco and a specially crafted plugin for Microsoft Internet Explorer (implemented as a CSP). On the server side, the implementation employs a modified version of AdNovum's Nevis Web portal. More specifically, it employs two Nevis components:

- A *secure reverse proxy* that is based on OpenSSL and Apache 2.0. OpenSSL is modified in the sense that it is possible to access and retrieve *Hash* from the SSL/TLS session cache.
- A Java- and CORBA-based *authentication service* that is integrated with a Vasco backend component (i.e., the Vacman controller). The authentication service gets a UAC from the client system and verifies it (by recomputing it from the server-side SSL/TLS hash value).

The general idea to make a C/R-based authentication token (such as a Vasco token) SSL/TLS session-aware is to replace the challenges that are otherwise pseudorandomly generated by the server (and transmitted to the browser) with values that are derived from the SSL/TLS session state (e.g., hash value). These values can be generated independently by the server and the browser, and if the two are connected through the same SSL/TLS session, then the resulting values will be the same. Our TLS-SA proof of concept implementation is also based on this general idea. A message sequence diagram for the protocol that is implemented is illustrated in Figure 1. It consists of the following steps:

1. The user requests a URL by entering it into his browser.

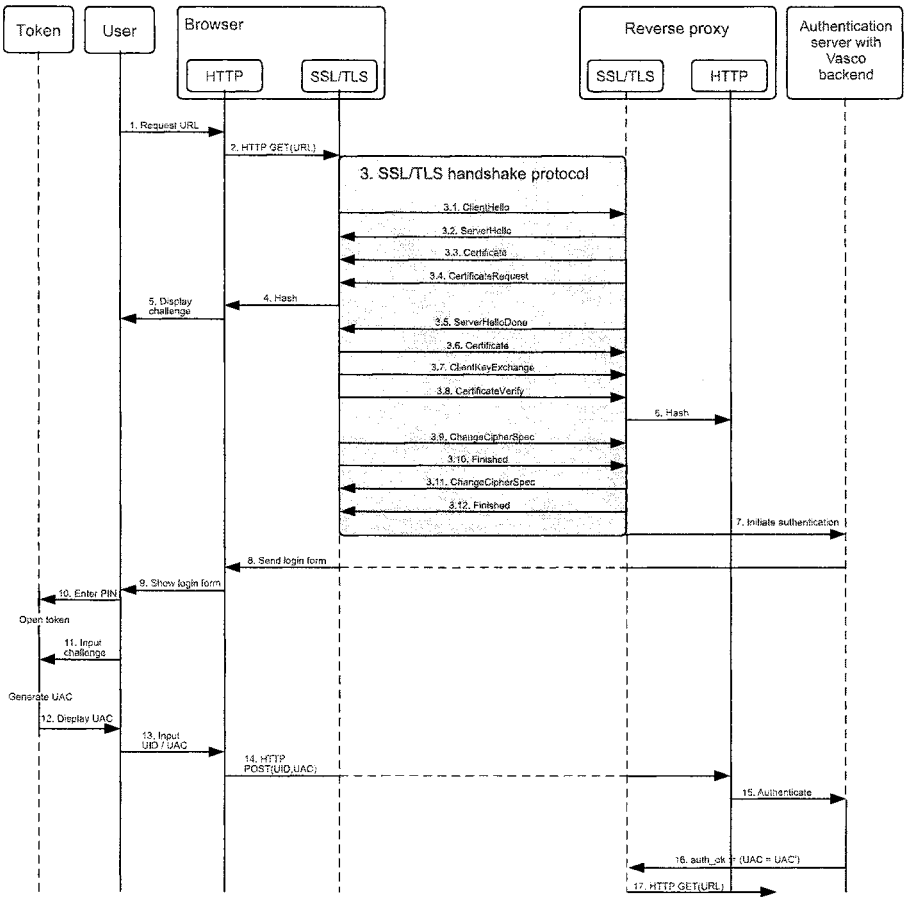


Figure 1 Message sequence diagram of our TLS-SA proof of concept implementation

2. The browser sends an HTTP GET request message for the URL to the Web server (or reverse proxy, respectively).
3. If the URL refers to a protected resource, then the Web server starts the SSL/TLS handshake protocol (according to the design principles of TLS-SA, we do not want the SSL/TLS handshake protocol to be changed). The server enforces certificate-based client authentication by sending out a CertificateRequest message to the browser in step 3.4. The browser, in turn, has a client certificate that is hardcoded (i.e., embedded) in the plugin. The certificate is sent to the server in step 3.6 as part of the Certificate message. In the CertificateVerify message of step 3.8, the browser proves knowledge of the private key. The bottom line is that the server can now be sure that the (anonymous) user employs the browser plugin and that the browser is bound to a particular SSL/TLS session (with a specific and bilaterally known

SSL/TLS hash value). For the protocol to work, it is not necessary that the private key remains secret, only its integrity is important or rather the entire plugin needs to be genuine.

4. The browser plugin grabs *Hash* from the SSL/TLS session state.
5. The browser plugin displays the challenge $Challenge = encode(Hash)$, i.e., a user-friendly encoding and representation of *Hash* to the user. Note that the encoding function *encode* is typically configurable by the user. Also note that the challenge is derived from the SSL/TLS session state, and that it is never sent from the server to the client. Because the Vasco token is only able to display decimal digits, the challenge must be encoded into this character set. This is achieved with a pseudorandom bit generator (PRBG) that is seeded with *Hash*. The output bit string is used to generate 4-bit sequences. Each 4-bit sequence is accepted if and only if it represents a decimal digit between 0 and 9 (otherwise, it is discarded). As soon as sufficiently many decimal digits are generated, the PRBG is stopped. In the current proof of concept implementation, the browser plugin can either display *Challenge* in a popup window or write it as a message to standard output.
6. On the server side, a filter in the reverse proxy server grabs *Hash* from the SSL/TLS session state and caches it together with the session ID. The server can trivially compute *Challenge* from *Hash* (using the *encode* function).
7. After having successfully executed the SSL/TLS handshake protocol, a secure channel between the browser and the reverse proxy is established. The reverse proxy checks whether the request is authenticated and forwards the unauthenticated request to the authentication service.
8. The authentication server sends a login form to the browser.
9. The browser displays the login form to the user.
10. The user enters his PIN into the token to unlock it.
11. The user enters the challenge as displayed in step 5 into the token.
12. The token calculates the response (i.e., the UAC) from the challenge and displays it to the user. The response comprises 7 decimal digits.
13. The user enters his user ID (UID) together with the UAC into the login form presented in step 9.
14. The browser sends the login form together with the user's credentials (i.e., UID and UAC) to the reverse proxy.
15. The reverse proxy forwards the user's credentials together with the challenge that was cached on the server side in step 6 to the authentication server.
16. The authentication server calculates the server-side response UAC' out of the server-side challenge and compares it with the client-side UAC sent by the client. Both responses must be identical in order to complete the authentication successfully. The authentication server sends back the result of the authentication (i.e., Boolean value `auth_ok`) to the reverse proxy.
17. If the authentication is successful (i.e., if `auth_ok` is `TRUE`), then the reverse proxy forwards the HTTP GET request message (as originally received in step 2) to the origin server.

Our proof of concept implementation demonstrates the basic idea of SSL/TLS session-aware user authentication and provides evidence of its feasibility. We point out and comment on several weaknesses and limitations next.

6 Weaknesses and Limitations

First, our proof of concept implementation has a weakness if the challenge is as short as described: the MITM can wait for a first SSL/TLS session to be established by the client. He can then set up a second SSL/TLS session to the origin server and modify the ServerHello message to be returned in the first session in a way that the deterministically compressed or truncated *Hash* value matches the *Hash* value of the second session. More specifically, the MITM looks for a second-preimage of *Hash* for the compression or truncation function in use. In our case, this requires only 5,000 guesses on the average until a second preimage is found (provided that the challenge is 4 decimal digits long). So few guesses could be explored before the expiration time of the challenge. If the MITM has found such a second preimage, he can spoof the user by simply forwarding the user's response to the origin server. The solution we see (but have not implemented so far) to remedy this weakness is to have the browser pseudorandomly select a few bits from the 36 bytes long hash value, and to form a challenge that consists of these bits and indexes to their positions within the hash value. The C/R device then encrypts this challenge in a way that can be decrypted by the server, and sends the resulting response to the server. This turns the server into an online validation authority that can limit the number of false guesses. The disadvantage of this solution is that the response must at least be as long as the block size of the block cipher (e.g., 64 or 128 bits).

Second, our proof of concept implementation is incomplete: we only provide support for one C/R token, one browser, and one server. The most critical part is the browser because it is unrealistic to assume that users will change their browser only due to security concerns. Consequently, we would like to see browser enhancements to support TLS-SA (e.g., [27]). Most importantly, we would like to see browsers that are able to display *Hash* or a user-friendly representation thereof (e.g., in the status bar). Enhancements like this can either be directly incorporated into a browser, or they can be implemented in the form of a browser plugin, as in our proof of concept implementation. It goes without saying that we prefer the first case. In the second case, the plugin must be professionally developed, integrated into a product, and deployed, all of which is nontrivial.

Third, TLS-SA and any implementation thereof does not protect against malware. If the client system hosts a Trojan horse or the browser is manipulated, then little can be done to protect the user.

7 Conclusions and Outlook

Many SSL/TLS-based e-commerce applications employ conventional user authentication mechanisms on the client side. It is not widely known, even within

the security community, that most of these mechanisms—if decoupled from SSL/TLS session establishment—are vulnerable to MITM attacks. Few institutions have fully recognized the seriousness of this threat and of those who have, many have declared personal certificates as the "strategic solution." To date, however, the often discussed obstacles to a full PKI rollout have prevented the successful, large-scale deployment of this solution.

Against this background, we think that TLS-SA provides a lightweight alternative to either a PKI[7] or the SSL/TLS extensions standardized by the IETF. Note that TLS-SA can work without modification of the SSL/TLS protocol. Also note that it is self-enforcing in the sense that users do not have to properly understand public key cryptography in order to be protected (*end user literacy*), and that the server can easily enforce the use of the protection mechanism (*enforcement and compliance*). This is in contrast to other protection mechanisms that rely on user education and proper browser configuration. Finally, note that TLS-SA has some privacy advantages, as it does not unambiguously identify the parties involved, also towards a illegitimately observing third party.

In this paper, we presented a proof of concept implementation of TLS-SA. This implementation is just one possibility to implement TLS-SA, and there are many other possibilities that may be explored. One particularly promising possibility is to make EMV-CAP[8] be SSL/TLS session-aware. In Europe, many financial institutions are providing their customers with EMV cards (to replace prior generations of payment cards). Hence, we expect most e-commerce users to possess one or more CAP-enabled EMV cards in the future. These cards are also intended for use in Internet banking. Under the conventional usage of SSL/TLS, however, they will be vulnerable to MITM attacks—with TLS-SA support they will not be.

References

1. Dierks T, Allen C: The TLS Protocol Version 1.0. RFC 2246, 1999.
2. Lopez J, Oppliger R, Pernul G: Why Have Public Key Infrastructures Failed so far? Internet Research, 15(5):544–556, 2005.
3. Mitchell J, Shmatikov V, Stern U: Finite-State Analysis of SSL 3.0. USENIX Security Symposium, 201–216, 1998.
4. Paulson LC: Inductive Analysis of the Internet Protocol TLS. ACM Trans. on Computer and System Security, 2(3):332–351, 1999.
5. Bleichenbacher D: Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. CRYPTO, 1–12, 1998.

[7] Of course the deployment of a PKI often has other objectives, e.g., the ability to provide nonrepudiation services.

[8] The EMV standard for smartcards was jointly developed and specified by Eurocard, MasterCard, and Visa. To use EMV cards for strong authentication in Internet-based e-commerce, MasterCard proposed the Chip Authentication Program (CAP) that was later adopted by Visa as Visa Passcode. It allows the user to generate OTPs or digital signatures.

6. Manger J: A Chosen Ciphertext Attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as Standardized in PKCS#1 v2.0. CRYPTO, 230–238, 2001.
7. Vaudenay S: Security Flaws Induced by CBC Padding—Applications to SSL, IPSEC, WTLS ... EUROCRYPT, 534–545, 2002.
8. Anderson RJ: Why Cryptosystems Fail. Communications of the ACM, 37(11):32–40, 1994.
9. Burkholder P: SSL Man-in-the-Middle Attacks. SANS Reading Room, 2002.
10. Oppliger R, Gajek S: Effective Protection Against Phishing and Web Spoofing. CMS, 32–41, 2005.
11. Desmedt Y, Goutier C, Bengio S: Special uses and abuses of the Fiat-Shamir passport protocol. CRYPTO, 16–20, 1987.
12. Fiat A, Shamir A: How To Prove Yourself: Practical Solutions to Identification and Signature Problems. CRYPTO, 186–194, 1986.
13. Cramer R, Damgård I: Fast and Secure Immunization Against Adaptive Man-in-the-Middle Impersonation. EUROCRYPT, 75–87, 1997.
14. Eronen P, Tschofenig H (Eds.): Pre-Shared Key Ciphersuites for Transport Layer Security (TLS). RFC 4279, 2005.
15. Badra M, Hajjeh I: Key-Exchange Authentication Using Shared Secrets. IEEE Computer, 39(3):58–66, 2006.
16. RSA Laboratories: OTP Methods for TLS. Draft 1, January 2006.
17. Steiner M., et al.: Secure Password-Based Cipher Suite for TLS. ACM Trans. Information and System Security, 4(2):134–157, 2001.
18. Taylor D, et al.: Using SRP for TLS Authentication. Work in progress, 2005.
19. Rivest RL, Shamir A: How to Expose an Eavesdropper. Communications of the ACM, 27(4):393–395, 1984.
20. Bellovin SM, Merritt M: An Attack on the Interlock Protocol When Used for Authentication. IEEE Trans. on Information Theory, 40(1), 1994.
21. Jakobsson M, Myers S: Stealth Attacks and Delayed Password Disclosure. 2005.
22. Kaliski B, Nyström M: Authentication: Risk vs. Readiness, Challenges & Solutions. BITS Protecting the Core Forum, October 6, 2004.
23. Asokan N, Niemi V, Nyberg K: Man-in-the-Middle in Tunneled Authentication Protocols. International Workshop on Security Protocols, 15–24, 2003.
24. Parno B, Kuo C, Perrig A: Phoolproof Phishing Prevention. Financial Cryptography, 2006.
25. Alkassar A, Stüble C, Sadeghi AR: Secure Object Identification—or: Solving The Chess Grandmaster Problem. Workshop on New Security Paradigms, 77–85, 2003.
26. Oppliger R, Hauser R, Basin D: SSL/TLS Session-Aware User Authentication—Or How to Effectively Thwart the Man-in-the-Middle. Computer Communications, 29(12):2238–2246, 2006.
27. Oppliger R, Hauser R, Basin D: Browser Enhancements to Support SSL/TLS Session-Aware User Authentication. W3C Workshop on Transparency and Usability of Web Authentication, 2006.

Secure TLS: Preventing DoS Attacks with Lower Layer Authentication

Lars Völker[1], Marcus Schöller[2]

[1] Institute of Telematics, Universität Karlsruhe (TH)
[2] Computing Department, Lancaster University

Abstract SSL/TLS has been designed to protect authenticity, integrity, and confidentiality. However, considering the possibility of TCP data injection, as described in [Wa04], it becomes obvious that this protocol is vulnerable to DoS attacks just because it is layered upon TCP. In this paper, we analyze DoS-attacks on SSL/TLS and describe a simple, yet effective way to provide protection for SSL/TLS by protecting the underlying TCP connection. We focus on a simple, feasible, and efficient solution, trying to balance security and usability issues by using the built-in key exchange of SSL/TLS to initialize TCP's MD5 option.

1 Introduction

In a world of e-commerce applications and Internet banking, users trust in their data—and, in consequence, their money—to be protected by state-of-the-art security protocols. Otherwise, lots of transactions would not take place. The dominant way of protection is to use the Secure Socket Layer (SSL) protocol [FFK98] or the closely related Transport Layer Security (TLS) protocol [DR06]. Many reasons exist for using TLS for such applications. (In the following, we will use the name TLS even though our ideas apply to the Secure Socket Layer (SSL) protocol in the same fashion.) The protocol has been widely deployed, it is easy to use, and applications can easily control the security configuration.

In recent years, so-called TLS-VPNs (virtual private networks) have become popular, using TLS connections to access resources of a central site. These solutions require long-lasting TLS connections to protect integrity and authenticity of packets. Considering injection attacks the choice of TLS implies that DoS attacks against these TLS-VPNs are possible. The simple solution of replacing TLS by another protocol is not feasible, as the simplicity and usability of TLS are still not offered by other solutions, as IPsec. While TLS itself may be as secure as possible, the use of TCP without additional cross-layer functionality makes it almost impossible to protect TLS against Denial-of-Service (DoS) attacks on the TCP layer. The author of [Wa04] calculated the time needed to successfully inject a packet in the stream and has shaped the understanding that injecting a packet into a TCP stream takes about 20 seconds in a realistic scenario (TCP window size: 65535, 1 Mbit connection, source port known). Naturally, the time depends on the speed of the communications links between the attacker and the

communication partners, the data rate they communicate with, and the TCP window size. The rationale behind the calculation is that a packet is successfully injected if IP addresses and port numbers are correct, and the sequence number fits into the targets window. With ever growing connection speed and distributed attackers the success rate and performance of a blind injection attack increase dramatically. Even worse, non-blind injection attacks, e.g. in wireless environments, are easier and can be achieved at any time in the TLS protocol if the attacker is able to read packets of just one of the two communications partners.

Solutions to stop injections attacks usually focus on stopping the injection of segments with special flags like the Reset-Flag—but not regular TCP segments. For these special segments, solutions like [SS06] demand its sequence number to resemble the very beginning of the window; thus, the packet is the very next packet expected. For regular data segments such behavior cannot be applied without loosing the advantages of TCP flow control; TCP connections would become very inefficient and slow. Unfortunately, for TLS the injection of just any data denotes a DoS attack. TCP assembles IP packets into a data stream and hands this stream over to TLS. In the effort of suppressing duplicate packets it hands over a given range of bytes just once. If an attacker is able to inject an arbitrary amount of random bytes into the byte stream TCP hands over to TLS, the data integrity check of TLS will fail, and the data has to be dropped. This behavior combined with TCP's dropping of duplicate packets leads to the TLS layer needing data that the TCP layer believes to have already handed over, and therefore detects as duplicates. A TLS implementation needs to restart the complete TLS connection in such a case.

In this paper, we investigate how to deal with the threat of DoS and describe the design, implementation, and the performance results of a solution to overcome theses problems. The solution we describe in the next section will add lower layer authentication to TLS in order to make packet injection impossible. TLS is used to exchange the keys and agree on the used cryptographic algorithm, which are used in an improved TCP MD5 option. We have added support of different cyrptographic algorithms (crypto agility).

In Section 3, we analyze what can be done to protect TLS and present our solution for securing TLS, while in Section 4, the implementation for Linux is described. Evaluation and results follow in Section 5, before we close with conclusion and outlook.

1.1 Related Work

In order to protect TLS one could use almost all existing mechanisms to protect TCP and/or IP. The IETF's TCP Maintenance and Minor Extensions (tcpm) working group has discussed many modifications to TCP to make blind injection attacks more difficult. One such example is [SS06]. In addition hot fixes in numerous operating systems exist, many of which could potentially reduce the impact of attacks by e.g. making port numbers and starting sequence numbers harder to be guessed. All these obviously work more or less well for segments

with special flags and a blind attacker. In all other cases, there is no protection at all.

A more general approach, protecting IP and upper layers could also be used. A good example is IPsec, whose current version [KS05] allows very strong protection of arbitrary data but fails to be useful for protecting connections to previously unknown peers. With an additional standard [RR05] it is possible to solve this problem but the needed DNS Security infrastructure [AAL+05] is not available today—and we do not assume this to change in the next years. In addition, the *Better-Than-Nothing Security* IETF working group started to research ways to protect every connection with IPsec, even if the regular level of IPsec security cannot be reached because validation of credentials is not possible. Protecting TLS with this envisioned solution would probably give enough security but with high overhead. Maybe it could replace TLS in the far future, but it seems more likely that just regular IPsec would be a better replacement than a *Better-Than-Nothing* solution.

Using TLS to protect data transported by UDP datagrams is the goal of Datagram TLS [RM06]. It adds sequence numbers to the TLS header and timers to implement the reliability needed by TLS. Since Datagram TLS implements reliability itself, injected packets can be easily detected and silently discarded without changing the state of the reliability mechanisms. The problems and solution presented in the next sections do not apply to Datagram TLS, which is not able to securely transport TCP segments.

2 Attacker Model and Requirements

Our analysis is based on two different attacker models. The *blind* attacker cannot read the packets between the two communications peers but can send packets with forged IP addresses. In addition the attacker could use techniques to improve the probability of guessing the correct source ports and initial sequence number [Za01].

The other attacker is *non-blind*. This could basically be another node on one of the peer's local networks or someone one the path between the peers. Especially, peers with a wireless connection or other broadcast networks are prone for non-blind attackers. These attackers are much more powerful than the blind attackers and have the additional capability to read packets sent by one of the communication peers to the other or the packets sent by both.

We propose the following requirements for securing TLS:

- Ability to cope with blind and non-blind attackers.
- Backward Compatibility to older TLS solutions.
- Minimal communication overhead.
- Minimal processing overhead.

While expecting that blind attackers is the more common threat for TCP connections, we did not want to exclude the non-blind attacker, which makes the protection of TLS much more difficult.

3 Problem Analysis and Solution Design

As we have seen, the major problem of TLS is that TCP offers a reliable service and delivers data to upper layer protocols just once because it is supposed to suppress duplicates. TCP's decision is based on the unsecured entries of the TCP header, or else it could find out that the received packet is a forgery. Even if TLS could figure out that a given data segment was injected, TCP would not pass TLS the original data since it seems to be duplicate. In addition to the generic ways of making TCP more DoS resistant, different specialized solutions for use with TLS could be envisioned:

1. Correction of TCP's state after injection
2. Retransmission on injection inside the TLS layer
3. Protection of TCP's messages

The *first possible solution* would insert cross-layer functions into TCP to *correct TCP's state after an injection has been detected.* TCP could wait for the next segment and pass it to TLS. Unfortunately, an attacker could inject more than one bogus segment. Assuming enough bandwidth, the attacker could send more packets in a given interval than the TLS instance can check in the same time. This would result in just another successful DoS attack. Furthermore, adding changes to the TCP implementation seems to be error prone and could introduce compatibility issues. Despite these problematic and complex changes, there are still attack scenarios—like non-blind data injections—this solution cannot deal with.

The second possible solution, making changes to TCP unnecessary and therefore increasing compatibilty, is the integration of retransmission mechanisms in the TLS layer. However, this would lead to a replication of TCP functionality by TLS.

The first two solutions have in common that the semantic of TLS has to be changed. The *third* solution tries keep the semantic intact: it tries to solve the problem by *adding lower layer authentication.* Possible transparent options are IPsec's Authentication Header (AH) [Ke05] and the TCP MD5 option. The major reason against AH is the still unsolved key exchange for previously unknown peers. Other reasons include the higher overhead in comparison to the MD5 option; using the same algorithm to protect integrity, a packet with AH would be 12 bytes larger than with MD5 option. With the MD5 option [MS95] the IETF has given users an option to protect their TCP connections and to mitigate the risk of data injection attacks. But the MD5 option has a major disadvantage: it does not include a mechanism to exchange the needed shared key. Using the MD5 option means setting up a shared key for the built-in algorithm manually. While this might be possible for long-living communication relations, like in routing, it is most certainly not an option for most web applications, like e-commerce, in which the client-to-server relation is often ad-hoc. Also, the MD5 option does not feature crypto agility, which means that just one algorithm (keyed MD5) can be used to protect messages. In the section 3.1 we present improvements to the MD5 option to overcome this problem.

Our proposal combines the advantages of TLS and the TCP MD5 option. TLS allows to setup a secure connection to an unknown peer but misses authentication of data below the TLS layer. The MD5 option adds the missing data authentication but cannot set up connections with unknown peers. That is why we aim at combining both approaches: we start TLS without protection first and use it to set up the MD5 option for the TLS connection as soon as possible. This significantly reduces the time window for an attack, especially when the receiver windows of TCP are very small at the start (injection less likely), like for example in Linux. Furthermore, many protocols reuse connections or use connections for a long time. HTTP 1.1 [FGM+99], for example, reuses TCP connections for later transfers, so setting up the protection is done seldom. The time during which an attack is possible is reduced to the first packets, namely the TCP three way handshake and the first phase of the TLS key exchange. In Section 5 we evaluate the attacker's window of opportunity in greater detail.

3.1 Improvements to the MD5 option

The TCP MD5 option is a popular way to protect BGP's TCP connections; it is not commonly applied for the protection of other connections. This is mainly due to three reasons:

1. Keys have to be set up manually because no key exchange exists.
2. MD5 is not considered to be secure anymore [WY05] and other cyptographic algorithms cannot be used.
3. The space available to TCP options is limited to 40 bytes [Po81], of which the MD5 option needs 18 bytes.

We overcome the *first* problem—no existing key exchange for the MD5 option—by using the key exchange of TLS. Additional overhead is not introduced.

Secondly, the security of the original algorithm used by the MD5 option seems to be too low for future usage. One could replace the algorithm with a more secure one, like HMAC-SHA-1-96 [MG98b] but as soon as SHA-1 is considered not strong enough, changes have to be done again. Therefore, we propose adding crypto agility to the MD5 option by allowing both parties to agree on the algorithm used in the option and recommend to keep MD5 as a fall back solution.

Table 1 shows proposed algorithms, in addition to the legacy algorithm (original MD5). The new entries are algorithms standardized for authentication of IPsec packets and can be easily added to the MD5 option.

The *third problem* is the limited space for TCP options. Usually the messages of the TCP three-way handshake transport several TCP options, like the Window Scaling option [JBB92]. This could have been a space problem with almost half the available space used for the MD5 option—however, these packets cannot be protected anyway since the TLS key exchange takes place afterwards. At this point in time, the MD5 option has not yet been negotiated. So this is of no

ID	Algorithm	Reference	Size incl. header
original MD5	original keyed MD5	[He98]	18 Bytes
HMAC-MD5-96	HMAC MD5, truncated to 96 bits	[MG98a]	14 Bytes
HMAC-SHA-1-96	HMAC SHA-1, truncated to 96 bits	[MG98b]	14 Bytes
AES-XCBC-MAC-96	AES XCBC, truncated to 96 bits	[FH03]	14 Bytes

Table 1. Proposed Algorithms for the MD5 Option

concern for our solution; usually, later segments transport less options, so space restrictions do not exist. An exception to this is the SACK option [MMFR96], which can use 10, 18, 26, or 34 bytes, depending on the number of blocks (1, 2, 3, or 4) to be transported. Often it is used in conjunction with the Timestamp option [JBB92], which uses 10 bytes. Using the 18 byte MD5 option would leave SACK with only two blocks or 1 block if the Timestamp option is needed too.

However, our proposed algorithms for the MD5 option can use less than the original 18 bytes of the option space, for instance 14 bytes if using HMAC with 96bit truncation[MG98a,MG98b]. This would already free 10% of the maximum option length. Further improvements might be able by truncating the hash even further or using future algorithms.

3.2 Changes to TLS

TLS key exchange and algorithm agreement is based on TLS cipher suites, each of which represents a valid combination of crypto algorithms. The TLS initiator will offer a list of supported algorithms and the responder will select one of them. Other options for communicating capabilities are not supported in TLS.

In order to let a TLS peer advertise the support of TCP MD5, new cipher suites must be added. We propose a new postfix to be appended to the cipher suites. A traditional cipher suite looks like: `TLS_DH_RSA_WITH_AES256_CBC_SHA` With our extension it would look like one of the following:

```
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA_USING_TCPOPTION_MD5
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA_USING_TCPOPTION_HMAC_MD5_96
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA_USING_TCPOPTION_HMAC_SHA1_96
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA_USING_TCPOPTION_HMAC_AES_XCBC_96
```

Backward compatibility to unmodified versions of SSL/TLS is automatically achieved by using the cipher suite concept because unknown cipher suites are ignored by the responder's TLS implementation.

3.3 Considering Network Address Translation

The original TCP MD5 option authenticates TCP's data and header, as well as the IP source address. If a NAT modifies the packet in transit the IP source address in the packet will be changed, and the receiver's integrity check with the TCP MD5 option will fail.

Different approaches can be taken:

- Communication of the original IP source address.
- Communication of the expected IP source address.
- Removal of IP source address authentication.

If the sender communicates its IP address a-priori, the receiver can adjust the integrity check accordingly, since it has the same data the sender used to calculate the hash in the first place. Likewise, the receiver could send the expected IP source address to the sender, so the sender can calculate the hash as expected by the receiver.

Since the communication of the IP addresses is not supported in TLS, we chose the third option: we recommend not to authenticate the sender's IP address. We already authenticate the sender as an entity using the TLS mechanisms, authenticating the IP address used is just a negligible improvement.

4 Implementation

Our implementation for Linux consists of three parts:

- *The MD5 option module* allows the protection of already existing connections. We also need to support additional algorithms since the MD5 option transports a 128bit keyed MD5 hash, which could be shortened. Furthermore, more secure algorithms exist, like HMAC-SHA-1.
- *The TLS implementation* exchange the cryptographic keys and parameters, which are passed to the MD5 option module to protect the TCP messages.
- *The interface* to connect the previous two components.

4.1 The MD5 option module

In current operating systems the MD5 option has rarely been implemented. In the Linux kernel it is even completely missing and might never be integrated [CMS02]. Therefore, we did not need to adapt an existing module but had the freedom of designing a module with our specific requirements. Our first approach was to implement the MD5 option as a Netfilter module—thus, having a modular system design. Unfortunately, this was not feasible because Linux already added TCP options before handing the packet over. Furthermore, it already calculated and adjusted the size of the TCP segment. As a consequence, there could have been not enough space left for the MD5 option. Therefore, we added the MD5 option processing directly into the Linux TCP/IP stack.

Operating systems which have already TCP MD5 support built in will be able to integrate our solution much faster, of course. These systems would only need to add an appropriate API for the communication with the MD5 option module in order to set up keys.

A flexible MD5 option module is favorable, especially if different algorithms can readily be added. The support of at least the original MD5 specifications is recommended for backward compatibility. In addition one stronger algorithm, HMAC-SHA-1-96, is recommended.

4.2 Modifications to the TLS daemon

We chose yaSSL as the TLS implementation to be used. The implementation had to be slightly modified to negotiate the new Cipher Suites, as presented in Section 3.2. The new Cipher Suite was added to the list of Cipher Suites, and an additional parameter was added to the object representing the negotiated Chipher Suite. The TLS implementation just needs to generate the keys for the chosen algorithm and can communicate these using the interface, described in Section 4.3.

4.3 An Interface to connect both Modules

Having support for generating and checking the MD5 headers, and having agreed on a certain algorithm and key material, only the connection between TLS and the TCP MD5 module is missing.

We could have used an existing interface, like IPsec's PF_KEY [MMP98]. For efficiency reasons we have chosen a socket-option-based interface for communication with the MD5 option component. Using the *setsockopts* call, the TLS implementation can send the data to the kernel which stores the data together with state of the socket. This is very efficient for the TCP MD5 option because the lookups of the needed options and keys can always be done in O(1) as soon as the packet is associated with its TCP connection.

The socket option transports several parameters:

- command (activate/deactivate)
- direction (incoming/outgoing)
- authentication algorithm
- cryptographic key
- key length

Using this parameter structure incoming and outgoing TCP MD5 options can be activated and deactivated independently. Using the authentication algorithm field, different algorithms are supported. Even different lengths of the needed keys can be adjusted. In addition a socket option to list all supported algorithms is useful.

5 Evaluation and Results

In Section 2 we outlined requirements for the TLS protection. We are confident that all these requirement are satisfied. We have considered both attacker models, allow for backward compatibility, and have only added a minimal communications overhead. The computational overhead is small and we will later present a modification to lower this overhead to an absolute minimum.

In order to allow for backward compatibility, our solution needs to start with the regular TLS handshake protocol, as shown in Figure 1. The first three packets form the three-way handshake of TCP. In the next step the ClientHello and

ServerHello messages are exchanged to agree on the Cipher Suite TLS should use: the ClientHello offers a list of supported Cipher Suites, one of which is chosen in the ServerHello message. We have added new Cipher Suites in order to signal the support for additional lower layer authentication and to allow for backward compatibility. A legacy client will not add a Cipher Suite with lower layer authentication and a legacy server would not select such a Cipher Suite, since he does not support it. For both the new mechanism is just an unknown Cipher Suite. The last messages include *KeyExchange*, to exchange the keys, *ChangeCipherSpec*, to signal that the protection should be started, and *Finished* to check that the previous messages have not been tampered with.

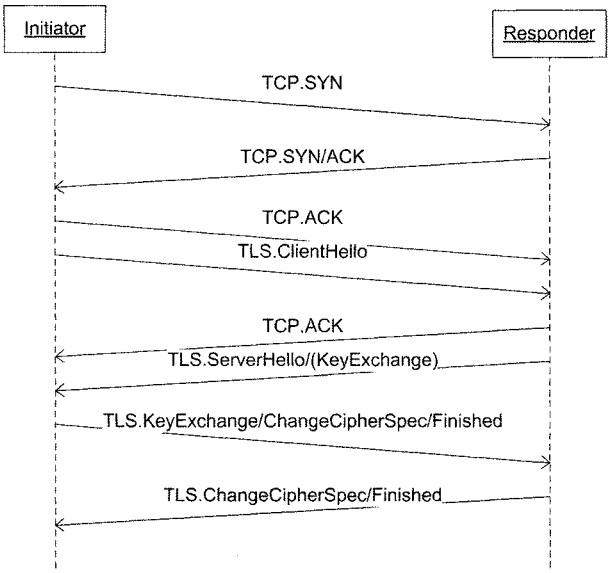


Figure 1. TLS start sequence

After these messages, our additional protection can be started, if the two peers negotiated it. Until this point in time 3 RTTs have passed—a short amount of time for the attacker. In the case of a TLS session being just resumed instead of setting it up again, the time will be 2.5 RTTs, since TLS needs one message less in this case.

An improvement of 0.5 to 1 RTT can be achieved by already protecting the last messages of the TLS handshake, but slightly changing the semantic of the TLS ChangeCipherSpec message: the message signals that the connection will be secured afterwards the improvement would already protect the integrity ealier.

In comparison with regular TLS we closed the attacker's window of opportunity to the TLS handshake phase—a compromise considering that backward

compatibility was required. We expect that further improvements cannot be done, since the TLS handshake is needed to negotiate the integrity protection algorithm first.

In the worst case, an open 65535 byte window and the attacker correctly guessing the source port, a *blind attacker* would need about 250ms at the rate of 90 Mbit/s [Wa04] to successfully inject a packet in the TCP stream. However, this scenario assumes a powerful attacker, who would also be capable of running a DoS attack by simply sending bogus packets at line speed. Therefore, no protection against such a DoS attack could be in place. After the TLS handshake phase, the attacker would not be able to successfully inject packets into the TCP stream. Therefore, the TLS connection is secure for the rest of its lifetime.

The evaluation result for the non-blind attacker is less perfect. The injection would not be possible after the handshake phase. During the TCP three-way handshake and the TLS handshake attacks are possible, and include attacks on the layer 2 medium, like deliberate frame collisions, attacks on ARP, attacks on the three-way handshake, attacks on the TLS Cipher Suite negotiation, and many more. Eliminating these attacks is close to impossible with current protocols. It is probably much easier to eliminate the non-blind attacker by securing the layer 2 medium. Overall, the time frame of possible DoS attacks has been closed significantly. As soon as the TLS handshake is finished, injections are not possible anymore.

We measured data throughput with and without the lower layer authentication by transferring 10 MB of data using TLS from a client to a local echo server and back. Using TLS MD5 authentication in addition increased the transmission time by 7.02% as in table 2. We expect higher penalties with stronger algorithms, like SHA-1.

min	avg	max	percentage	cipher suite
3.67	3.6819	3.72	65,22%	DHE_RSA_AES256_NULL
5.62	5.645	5.69	100,00%	DHE_RSA_AES256_MD5
4.07	4.0887	4.13	72,43%	DHE_RSA_AES256_NULL_TCP_MD5
6.02	6.0412	6.11	107,02%	DHE_RSA_AES256_MD5_TCP_MD5

Table 2. Performance Results

While 7.02% seem to be a significant cost, we have left an option of improvement. The system presented actually protects the integrity of the data twice: before the encryption using TLS mechanisms and afterwards by using the MD5 option. Another 100 runs using just the TCP-MD5 authentication were even faster than using only the regular TLS MD5 authentication by 27.60%. This surprising value seems to be the result of a much better tuned Linux Kernel MD5 implementation and not so much of the performance difference between HMAC-MD5 and keyed MD5. Since the loopback device was used for measurement the results are limited by the processing speed and not the network. Re-

ducing the size of data by using HMAC in the lower layer authentication and no TLS authentication could improve overhead, and transmission speed.

6 Conclusion and Outlook

We have presented a simple, yet effective solution to protect SSL and TLS from DoS attacks introduced by TCP weaknesses. Instead of changing TCP to make attacks on it harder, we have shown that a better way to protect SSL and TLS exists: using an integrity check option of TCP.

While using the TCP MD5 option header allows for much better security compared to regular TLS solutions, future improvements can be easily introduced by adding new algorithms to generate and check the integrity values transported in the MD5 option headers. Keyed MD5 is not a state-of-the-art solution, while HMAC-SHA-1 truncated to 96 bits is better suited to protect the integrity of the data and needs to transfer only 14 bytes instead of 18. But even stronger algorithms could be specified for future usage.

Further improvements to make TLS more DoS resistant are still possible. An attacker who replays one packet at a very fast rate, could induce a high load on the MD5 option module, no matter if the packet's sequence number fits the window or not, because the sequence number is checked after the MD5 option's integrity check. The receiver's processing speed of the integrity algorithm would be a performance bottleneck the attacker could exploit for a DoS attack. An improvement to the TCP MD5 option would be checking the sequence number before checking the integrity check value. This would allow to discard replayed packets, which would have been discarded after the integrity check anyway.

A downside to our solution is the additional usage of the limited TCP option space. Future work will include checking if this negatively effects the performance of TCP and if shorter authentication schemes are possible. An evaluation is necessary to show if HMAC-SHA-1 truncated to 64 bits or less could still be strong enough for short living connections. It is possible that enough protection against DoS attacks is offered, while leaving more TCP option space to SACK and other options.

Acknowledgment

The authors would like to thank Christoph Sorge and Erik-Oliver Blaß for inspiring discussions, and Stefan Cyris for help with the implementation.

References

[AAL+05] Arends, R., Austein, R., Larson, M., Massey, D., und Rose, S. DNS Security Introduction and Requirements. RFC 4033. March 2005.

[CMS02] Cox, A., Miller, D. S., und Schwartz, D. *RFC2385 (MD5 signature in TCP packets) support.* Linux Kernel Mailinglist. Mar 2002.

[DR06] Dierks, T. und Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346. April 2006. Updated by RFCs 4366, 4680, 4681.

[EJ01] Eastlake 3rd, D. und Jones, P. US Secure Hash Algorithm 1 (SHA1). RFC 3174. September 2001.

[FFK98] Freier, A. O., Freier, A. O., und Kocher, P. C. The SSL Protocol Version 3.0. Draft. Nov 1998. `http://wp.netscape.com/eng/ssl3/`.

[FGM+99] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., und Berners-Lee, T. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616. June 1999. Updated by RFC 2817.

[FH03] Frankel, S. und Herbert, H. The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec. RFC 3566. September 2003.

[He98] Heffernan, A. Protection of BGP Sessions via the TCP MD5 Signature Option. RFC 2385. August 1998.

[JBB92] Jacobson, V., Braden, R., und Borman, D. TCP Extensions for High Performance. RFC 1323. May 1992.

[Ka05] Kaufman, C. Internet Key Exchange (IKEv2) Protocol. RFC 4306. December 2005.

[KBC97] Krawczyk, H., Bellare, M., und Canetti, R. HMAC: Keyed-Hashing for Message Authentication. RFC 2104. February 1997.

[Ke05] Kent, S. IP Authentication Header. RFC 4302. December 2005.

[KS05] Kent, S. und Seo, K. Security Architecture for the Internet Protocol. RFC 4301. December 2005.

[MG98a] Madson, C. und Glenn, R. The Use of HMAC-MD5-96 within ESP and AH. RFC 2403 (Proposed Standard). November 1998.

[MG98b] Madson, C. und Glenn, R. The Use of HMAC-SHA-1-96 within ESP and AH. RFC 2404. November 1998.

[MMFR96] Mathis, M., Mahdavi, J., Floyd, S., und Romanow, A. TCP Selective Acknowledgement Options. RFC 2018. October 1996.

[MMP98] McDonald, D., Metz, C., und Phan, B. PF_KEY Key Management API, Version 2. RFC 2367. July 1998.

[MS95] Metzger, P. und Simpson, W. IP Authentication using Keyed MD5. RFC 1828. August 1995.

[NG] Netfilter-Group. The netfilter/iptables project homepage. Website. `http://www.netfilter.org/`.

[Po81] Postel, J. Transmission Control Protocol. RFC 793. September 1981. Updated by RFC 3168.

[Ri92] Rivest, R. The MD5 Message-Digest Algorithm. RFC 1321 (Informational). April 1992.

[RM06] Rescorla, E. und Modadugu, N. Datagram Transport Layer Security. RFC 4347. April 2006.

[RR05] Richardson, M. und Redelmeier, D. Opportunistic Encryption using the Internet Key Exchange (IKE). RFC 4322. December 2005.

[SS06] Stewart, R. und Stewart, R. Improving TCP's Robustness to Blind In-Window Attacks. Draft v5. Jun 2006.

[Wa04] Watson, P. A.: Slipping in the Windows: TCP reset attacks. In: *Cansecwest.* 2004.

[WY05] Wang, X. und Yu, H.: How to break md5 and other hash functions. In: *Advances in Cryptology - Eurocrypt.* 2005.

[Za01] Zalewski, M. Strange Attractors and TCP/IP Sequence Number Analysis. Whitepaper. Apr 2001. `http://www.bindview.com/Services/Razor/Papers/2001/tcpseq.cfm`.

Teil V

Preisträger

Traffic Shaping in a Traffic Engineering Context

Dirk Abendroth

BMW AG, Research and Innovation Center, Hufelandstr. 8a, 80788 Munich, Germany

Abstract. Modern communication networks carry a broad range of various applications such as data, voice, and video, and network providers are increasingly faced to the problems of traffic and resource management in order to meet the increasing quality of service (QoS) requirements. In particular, multi-media and realtime applications are sensitive with respect to QoS. In this paper we propose novel traffic engineering concepts for QoS that, in contrast to already existing solutions, need no intelligent functionality inside the core network and furthermore provide access to resource dimensioning in a comfortable way.

1 Introduction

The problems of resource dimensioning and providing QoS are obviously linked directly to each other and are aggregated by the fact that the traffic entering the network considered in general is unpredictable in both, its intensity and its characteristics. One way to make traffic flows predictable is to use traffic shapers which regulate passing flows to certain pre-defined traffic characteristics. One weak point of today's standard shaping algorithms (Leaky Bucket and Token Bucket) is the fact, that shaped flows have to be reshaped to enforce conformance at each node on their path and that shaping delays and hardware costs add up hop-by-hop. This gave reason to develop traffic engineering approaches based on shaping algorithms that enforce the passing streams at network edges to a certain "natural" characteristic which 1) is carried throughout the network without re-shaping, 2) needs only very few describing parameters, and 3) leads to straight forward resource dimensioning rules.

Based on the (large deviations) effective bandwidths we present the following approach: effective bandwidths (EB) behave additively for independent multiplexed streams and do not change when passing a network node (in the many sources limiting regime). The introduced "Effective Bandwith Shapers" (EBSs) upper limit the EB of a departing flow to a pre-defined threshold value. Similar as above, by use of these Effective Bandwidths Shapers the network can be dimensioned due to simple rules derived from the large deviations theory and QoS guarantees can be provided in a comfortable way.

2 Effective Bandwidth Traffic Shapers

Effective bandwidths are a statistical descriptor for an arrival process sharing a resource with other flows. The effective bandwidth captures the effect of the considered flow's rate fluctuation and burstiness and represents the rate (or bandwidth) at which the considered flow 'effectively' needs to be served according to certain quality of

service demands. The easy handling of effective bandwidths is mostly up to its additivity, i.e. the effective bandwidth of a flow consisting of two independent flows merged together is simply the sum of the two independent single flows' effective bandwidths [1]. Inductively this property applies also to each number of merged flows larger than two, of course. Apart from that, [2] was able to show analytically, that under the many sources limiting regime, i.e. the number of independent inputs to the switch increases and the buffer size per input and the service rate per input stay fixed, the following property holds: the large deviations characteristics and especially the effective bandwidth assigned to a flow does not change when passing a switch. Inductively we draw the conclusion that an effective bandwidth initially assigned to a flow keeps being the same on the flow's entire path through the network. Given the routing information is known, the effective bandwidth of all sub and super flows can be determined easily and resource dimensioning can be done based on simple rules derived from the large deviations theory. The problem with the above scheme is that the flows' effective bandwidths generally are not known. This is due to the fact that a source's future behavior depends on the user and the application. In case of a feedback-based network protocol (e.g. transmission control protocol, TCP) the effective bandwidth also strongly depends on the load situation in the network, and thereby on the other users' behavior, which is hardly known as well. We suggest a new shaping and policing scheme that limits the effective bandwidth of passing streams to a predefined value and intervenes only if necessary. The departing flows are characterized by a (known) upper bound on the effective bandwidth which, of course, behave additively as well and are valid throughout the entire network corresponding to the above argumentation.

We address the question how to limit effective bandwidths of passing streams at a minimum delay. Furthermore we proof the effective bandwidths' 'insensitivity to switches' at the presence of different TCP protocols and shown that the invariance of effective bandwidths (derived under the assumption of the many sources limiting regime) holds approximately already for a small number of flows sharing a network node. Finally we investigate the interaction between the introduced effective bandwidth based shaping and policing algorithm and TCP's closed loop control with respect to throughput and delays.

2.1 Definition, Interpretation, and Estimation of Effective Bandwidths

In this section we briefly recall the definition and interpretation of effective bandwidths and introduce the effective bandwidth estimation algorithm our shaping scheme relies on.

The effective bandwidth of a stationary flow has been defined [3] as

$$\alpha(\Theta,t)=\frac{1}{\Theta t}\log\left(\mathrm{E}\left\{e^{\Theta X[\tau,\tau+t]}\right\}\right) \tag{1}$$

depending upon the space parameter Θ and the time parameter t. $X[\tau,\tau+t]$ denotes the amount of data arriving in the time interval $[\tau,\tau+t]$. The effective bandwidth assigned to a flow is upper bounded by the flow's peak rate and lower bounded be the flow's

mean rate. If aflow has only little rate fluctuations the effective bandwidth is close to the mean rate whereas bursty streams have effective bandwidths significantly larger than the mean rate.

For real data traces of finite length the above definition needs to be approximated be an appropriate effective bandwidth estimator. Among others (e.g. Kullback-Leibler Distance (KLD) estimator, Linear Regression (LR) estimator, [4]) exists the 'direct effective bandwidth estimator' [5] that simply replaces the probabilistic expectation value in (1) with according time average,

$$\tilde{\alpha}_{direct}(\Theta,t) = \frac{1}{\Theta t}\log\left(\frac{1}{\mathrm{T-t}}\int_{0}^{\mathrm{T-t}} e^{\Theta X[\tau,\tau+t]}d\tau\right) \tag{2}$$

The finite trace length is taken care of by the parameter T. The direct estimator requires no restrictions on the traffic flow apart from ergodicity (expectation values coincide with time averages.

Our shaping scheme, of course, needs to estimate the departing flow's effective bandwidth and is based on the 'block estimator'. The block estimator operates similar to the direct estimator but considers non-overlapping (time) blocks of length t, [6]:

$$\tilde{\alpha}_{block}(\Theta,t) = \frac{1}{\Theta t}\log\left(\frac{1}{\lfloor \mathrm{T/t} \rfloor}\sum_{i=1}^{\lfloor \mathrm{T/t} \rfloor} e^{\Theta X[(i-1)t,it]}\right) \tag{3}$$

The block estimator assumes that the number of arrivals per block are realizations of independent and identically distributed (i.i.d.) random variables and its applicability is thus limited to short-range dependent streams.

2.2 The Shaping/Policing Algorithm

All shaping algorithms mentioned in the previous section operate off-line, i.e. the entire trace needs to be known prior to estimation. For an interacting shaping and policing algorithm an on-line effective bandwidth estimation is mandatory. Our shaping algorithm employs the on-line 'recursive block estimator' (4), (5) which is equivalent to the block estimator (3), cf. [7]: M_k, initially set to zero, is recalculated each time block recursively by

$$M_{k+1} = \left(1-\frac{1}{k}\right)M_k + \frac{1}{k}e^{\Theta X[(k-1)t,kt]},\ M_0 = 0 \tag{4}$$

until the recursion fulfills a given condition, e.g. if Mk varies by no more than a given tolerance, or in case of a finite trace if the trace's end is reached. The final M value obtained leads to the estimated (empirical) effective bandwidth

$$\tilde{\alpha}_{recursive}(\Theta,t) = \frac{1}{\Theta t}\log\left(M_{final}\right) \tag{5}$$

Having specified the effective bandwidth estimator we turn towards the shaping algorithm itself. The algorithm aims to emit data flows that do not exceed a pre-defined effective bandwidth threshold value EBth. At the same time the introduced shaping delays shall be reduced to a minimum. The shaper's philosophy is as follows. As long as the incoming stream has no larger effective bandwidth than EBth, the shaper is transparent. If the effective bandwidth value of the incoming stream exceeds EBth, the shaper intervenes such that the departing flow is assigned an effective bandwidth of EBth. The algorithm works as follows: In the setup phase, the algorithm translates the given upper bound EBth (in [bit/s]) into an equivalent *upper bound on* M, i.e. with $\tilde{\alpha}_{recursive}(\Theta,t) = EB_{th}$ we rearrange (5):

$$M^{max} = \exp\left(\frac{EB_{th} \cdot \Theta t}{8\frac{bit}{byte} \cdot spu}\right) \tag{6}$$

The space parameter unit (spu) is set to an Ethernet packet size of 1500 bytes and is used (in combination with the factor 8 bit/byte) for translation of the effective bandwidth from [bit/byte] into space parameter units.

The *block size*, i.e. the *time parameter* t, is proposed to be set to the mean transmission time of one spu-sized packet, [8]. Different from UDP, the mean rate of TCP connections is not known prior to submission and might vary over a wide range even if the associated effective bandwidth stays constant. One possibility is to estimate the mean rate $\bar{\rho}$ of the incoming flow and to determine the time parameter as

$$t = \frac{1[spu]}{\min\{\bar{\rho}, EB_{th}\}\left[\frac{spu}{s}\right]} = \frac{8\left[\frac{bit}{byte}\right] \cdot spu\,[bytes]}{\min\{\bar{\rho}, EB_{th}\}\left[\frac{bit}{s}\right]} \tag{7}$$

The mean rate estimation employed might be for instance the iterative algorithm proposed in [9]:

$$\bar{\rho}_{new} = \left[\left(1 - e^{-\frac{\Delta t}{c}}\right) \cdot \frac{s}{\Delta t}\right] + e^{-\frac{\Delta t}{c}} \cdot \bar{\rho}_{old} \tag{8}$$

where s denotes the current packet's size, Δt is the current inter-arrival time, and c is a scaling time constant. A second possibility is to simply use the effective bandwidth threshold value EBth instead of the mean rate

$$t = \frac{8\left[\frac{bit}{byte}\right] \cdot spu\,[bytes]}{EB_{th}\left[\frac{bit}{s}\right]} \tag{9}$$

i.e. to set the block size to the 'effective transmission time' of one spu-sized packet (which is smaller than the mean transmission time). Simulation test runs show that the latter solution outperforms the former rate estimation method in terms of a 'gain versus complexity' trade-off by far (notice that the variable mean rate variant also requires an Mmax update each block). In the sequel we will use the constant time parameter according to (9) only.

In order to limit the effective bandwidth of the passing stream the shaping algorithm calculates the maximum number (or rational fraction) of spu-sized packets that may pass the shaper during this block such that the effective bandwidth of the departing flow does not exceed EBth, cf. (4):

$$X_{k+1}^{\max} = \max\left\{0, \frac{1}{\Theta}\log\left[kM_{\max} - (k-1)M_k\right]\right\} \tag{10}$$

M performs an update each block according to (4). The shaping decision whether to delay or pass a packet is derived from a scalar comparison of the number of packets already sent in this block X_k to the upper bound X_k^{max}. The above defined algorithm shall be called effective bandwidth shaper with counter implementation, EBS^{count}.

2.3 Simulation Results

We consider a chain with several hops, cross traffic, and effective bandwidth shapers. Data sources and sinks communicate in all scenarios via TCP. We used the discrete event domain of the Ptolemy simulation environment with full TCP implementations including connection establishment and termination. The simulation time of each simulation run was set to 1800s, i.e. 30 min. real time.

We will demonstrates in this section that the EBS^{count} algorithm works properly when applied to reactive TCP traffic. Furthermore we will shown that the invariance of effective bandwidths (derived under the assumption of the many sources limiting regime) holds approximately already for a small number of flows sharing a network node.

We consider a chain with several hops and add-drop cross traffic as depicted in Fig. 1. As foreground traffic a number of 10 TCP sources (S1 .. S10) communicates to its dedicated destinations (D1 .. D10) via several hops (routers R1 .. R5). Each hop the foreground traffic competes with 10 add-drop TCP connections (cross traffic, X1 .. X10 etc.). All connections use TCP NewReno and each connection (foreground and background) is limited to an effective bandwidth threshold value of 1 Mbit/s by means of an EBScount. Since the results we obtained from different space parameters have been similar, we restrict ourselves to a space parameter of Θ =0.5 only.

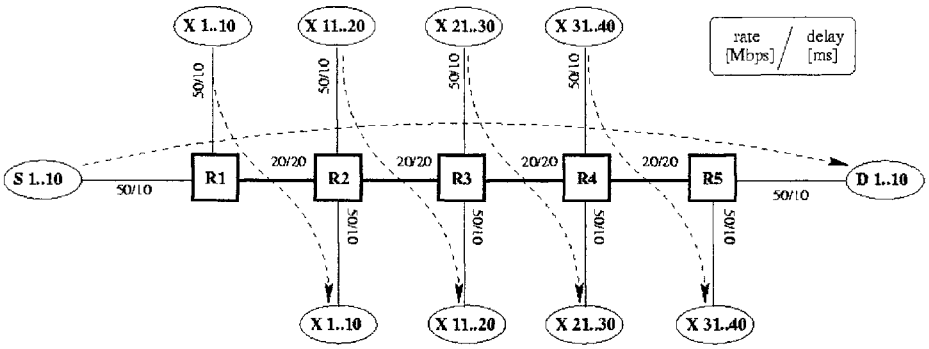


Figure 1 Scenario: TCP chain with TCP cross traffic, shaped flows.

The bottleneck routers' buffers are limited according to the bandwidth-delay product

$$B_{R_i} = rnd\left(\frac{20\,Mbit/s \cdot 20ms}{8\,bit/byte \cdot 1500\,byte/packet}\right) = 33 packets, \tag{11}$$

whereas all shaper buffers have a capacity of 10 packets. We study how the foreground flows' effective bandwidths change while passing the hops. Figure 2 (left) is a plot of the foreground flows' effective bandwidths behind the routers R1 .. R5 (hop 1 .. hop 5) compared and normalized to the corresponding effective bandwidths before entering router R1 (hop 0),

$$\varepsilon_{EB}^{hop\ j} = \left|\frac{EB^{hop\ j} - EB^{hop\ 0}}{EB^{hop\ 0}}\right|. \tag{12}$$

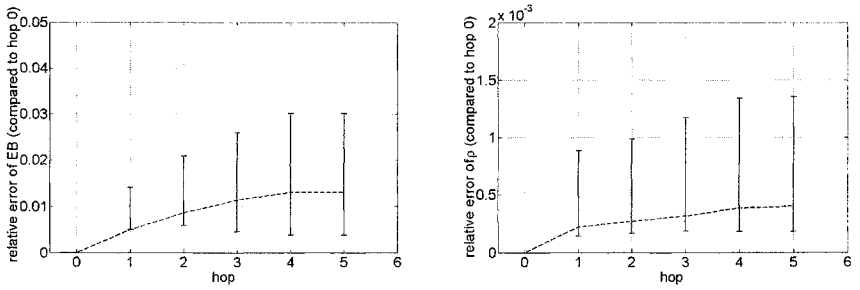


Figure 2 Relative errors of effective bandwidth (left) and mean rate (right), foreground traffic, versus hop: minimum, mean, and maximum.

The dashed line connects the mean error values whereas the error intervals mark minimum and maximum errors. As expected the mean errors increase along the hops

but stagnate at less than 1.5% at hop 4 and hop 5. The minimum-maximum intervals also enlarge with increasing hop number but similar to the mean values meet a maximum size already at hop 4 and do not increase from hop 4 to hop 5. The maximum error is also stabilized at around 3%.

Each hop the flows might experience losses due to the limited router buffer sizes. Consequently, the flows' mean rates offered to the subsequent routers decrease monotonic. As to see the influence of the decrease of the mean rates on the errors of the effective bandwidths figure 2 (right) provides a similar plot for the flow's mean rates (MRs): foreground mean rates behind routers R1 .. R5 (hop 1 .. hop 5) compared and normalized to the corresponding mean rates before entering router R1 (hop 0),

$$\varepsilon_{MR}^{hop\ j} = \left| \frac{MR^{hop\ j} - MR^{hop\ 0}}{MR^{hop\ 0}} \right| \tag{13}$$

The general shape is similar to that of Fig. 2 (left) but minimum, mean, and maximum change (loss) are more than one order of magnitude smaller compared to the effective bandwidths, hence, in comparison negligibly small.

Figure 3 explicitely shows the absolute decrease of the foreground traffic's mean rate along the five bottleneck routers due to losses at finite router buffers.

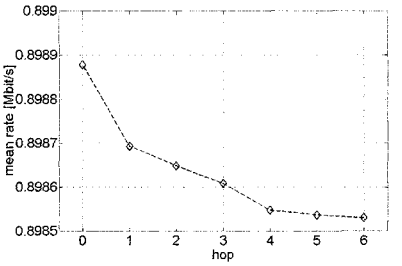


Figure 3 Mean rate (foreground traffic) versus hop.

The ratio between mean rate and effective bandwidth (threshold value) is roughly speaking 0.899. The link utilization was slightly above 90% due to the fact that the (in terms of round trip times) short cross traffic connections are able to get a slightly higher mean rate than the foreground traffic flows which communicate over five hops.

Summarizing, the effective bandwidths of the foreground traffic stay approximately constant with average error smaller than 1.5% and maximum error of 3%. At the same time the change of mean rates (due to losses) behind the shapers is less than 0.05% over five hops at a link load of more than 90%.

3 Summary

We introduced a traffic engineering approach based on shaping algorithms that enforce the passing streams at network edges to a certain "natural" characteristic which is carried throughout the network without re-shaping, needs only very few describing parameters, and leads to straight forward resource dimensioning rules.

Effective bandwidths behave additively for independent multiplexed streams and do not change when passing a network node. The introduced Effective Bandwith Shapers upper limit the EB of a departing flow to a pre-defined threshold value. The invariance of effective bandwidths to network nodes is shown to hold already for a small number of competing flows in a TCP szenario.

4 References

1. C.-S. Chang, and J.A. Thomas: *Effective bandwidth in high speed digital networks*, IEEE Journal on Selected Areas in Communications, vol. 13, no. 6, pp 1091-1100, August 1995.
2. D.Wischik: *The output of a switch, or, effective bandwidths for networks*, Queueing Systems, Volume 32, pp 383-396, 1999.
3. F.P. Kelly: *Notes on effective bandwidths*, In F. P. Kelly et al., "Stochastic Networks: Theory and Applications", pp 141-168. Oxford University Press, 1996.
4. S. Tartarelli, M. Falkner, M. Devetsikiotis, I. Lambadaris, and S. Giordano: *Empirical effective bandwidths*, Proceedings IEEE GLOBECOM 2000, vol. 1, pp 672-678, 2000.
5. R.J. Gibbens: *Traffic characterisation and effective bandwidths for broadband network traces*, Statistical Laboratory Research Report 1996-9, University of Cambridge, 1996.
6. N.G. Duffield, J.T. Lewis, N. O'Conell, R. Russell, and F. Toomey: *Entropy of ATM streams: a tool for estimating QoS parameters*, IEEE Journal on Selected Areas in Communications, vol. 13, no. 6, pp 981-990, August 1995.
7. J. Yang, and M. Devetsikiotis: *On-line estimation, network design and performance analysis*, Proceedings ITC 17, Elsevier Science, 2001.
8. D. Abendroth, and U. Killat: *Effective bandwidth shaping: a framework for resource dimensioning*, Proceedings IEEE ICON 2003, Sydney, Australia, 2003.
9. O. Bonaventure, and S. De Cnodder: *Request for Comments: 2963*, category: informational, October, 2000.

Routing and Broadcasting in Ad-Hoc Networks

Marc Heissenbüttel

University of Bern, Switzerland
heissen@iam.unibe.ch

Abstract. In this paper, we introduce two protocols - a routing and a broadcasting protocol - for ad-hoc networks which are based on a new paradigm enabled by the broadcast property of the wireless propagation medium. Nodes simply broadcast packets such that forwarding decisions are no longer taken at the sender of a packet, but in a completely distributed manner at the receivers. Consequently, nodes do not require knowledge about their neighbors. In this way, control traffic can be eliminated almost completely which in turn conserves scarce network resources such as battery power and bandwidth. Furthermore, as these two protocols are almost stateless and nodes do not store network topology information they remain unaffected by even very high rates of topology change and prove highly scalable in terms of number of nodes.

1 Introduction

Ad-hoc networks consist of a collection of wireless hosts that operate without the support of any fixed infrastructure or centralized administration and are completely self-organizing and self-configuring. Nodes are connected dynamically and in an arbitrary manner to form a network, depending on their transmission ranges and positions. Network operations like routing and broadcasting are difficult tasks in such a dynamic environment and have been subject of extensive research over the past years.

Many such protocols designed for ad-hoc networks have been proposed in the literature. Basically, we can distinguish for both, routing and broadcasting protocols, between topology-based and position-based protocols. Overviews can be found in [1], [2], [3]. Like protocols in the Internet, topology-based routing protocols use routing tables and information about available links to forward packets based on the destination address. On the other hand, in position-based protocols (also known as geographical, geometric, or location-based routing protocols), the nodes' geographical positions are used to make forwarding decisions. Therefore, a node must be able to determine its own position and the position of the destination node. This information is generally provided by a global navigation satellite system and a location service [4], respectively. Furthermore, nodes obtain knowledge of their neighbors through beacons, short hello messages broadcasted periodically from each node.

In this paper, we present a position-based routing protocol for ad-hoc networks, called Beacon-Less Routing Protocol (BLR) [5] that allows nodes to route

packets without having information about their neighboring nodes by introducing a concept of Dynamic Forwarding Delay (DFD). BLR is based on a new routing paradigm enabled by the broadcast property of the wireless propagation medium. Forwarding decisions are not taken at the sender of a packet, but in a completely distributed manner at the receivers and are solely based on the position of the destination and the receiving node itself. We also present the Dynamic Delayed Broadcasting Protocol (DDB) [6] that uses the same concept DFD, which allows locally optimal broadcasting without any prior knowledge of the neighborhood.

The remainder of this paper is organized as follows. In Section 2 and Section 3, we introduce the Beacon-Less Routing protocol (BLR) and the Dynamic Delayed Broadcasting Protocol (DDB), respectively, and provide some analytical and simulation results. Section 4 concludes the paper.

2 The Beacon-Less Routing Protocol (BLR)

2.1 Protocol Description

Like any other position-based routing algorithms, we assume that nodes are aware of their own positions and that the source has the possibility to locate the position of the destination node. However, as the fundamental difference to other position-based routing algorithms, nodes do not require information about their neighboring nodes, neither about their positions nor even about their existence.

BLR has two main modes of operations. BLR routes packets in greedy mode whenever possible. If greedy routing fails, BLR switches to backup mode to recover and route the packet further.

In greedy mode, a node that has a packet to forward simply broadcasts it. Consequently, all neighbors receive the broadcast packet. The protocol ensures that just one of the receiving nodes relays the packet further. This is accomplished by different forwarding delays at different receiving nodes and restricting the nodes that are allowed to forward the packet to a certain area, called forwarding area. Nodes within this area can mutually receive each others transmissions. For the forwarding area BLR uses a circle with diameter r relative to the forwarding node S in the direction of the final destination D as depicted in Fig. 1. A receiving node can determine if it is within the forwarding area from its own position and the positions of the destination D and the previous node S, which are both stored in the packet header. Potential forwarders, e.g., A and B in Fig 1, calculate a Dynamic Forwarding Delay (DFD) in the interval $[0, Max_Delay]$ depending on their position relative to the previous and the destination node. The DFD is calculated by (1) with r as the transmission radius of a node, p the node's progress towards the destination, and Max_Delay as a system parameter that indicates the maximum time a packet can be delayed per hop. Nodes outside the forwarding area simply drop the packet (node C).

$$Add_Delay = Max_Delay \cdot \left(\frac{r - p}{r} \right) \tag{1}$$

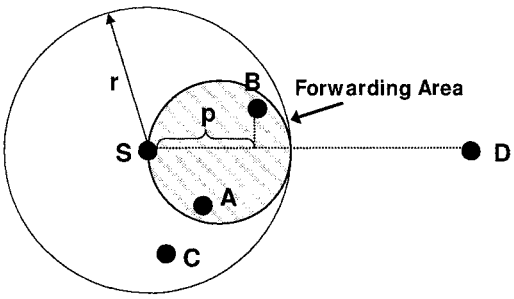


Fig. 1. Forwarding Area with potential forwarders A and B

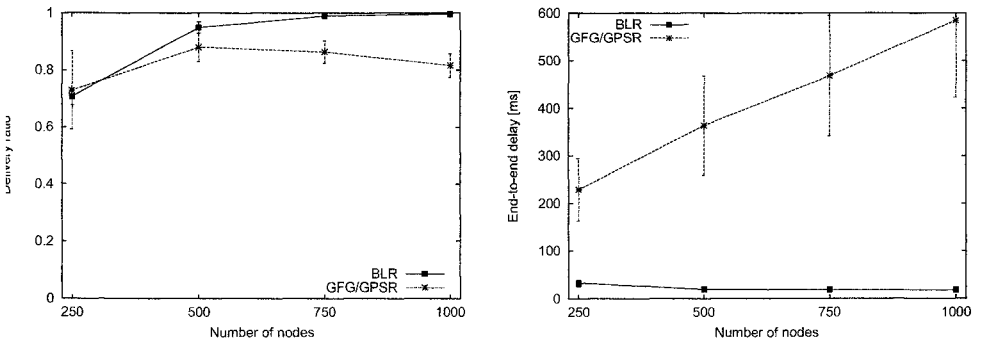
According to this DFD function, the node with the most progress (e.g., node B), i.e., closest to the destination, calculates the shortest *Add_Delay* and thus rebroadcasts the packet first. This minimizes the number of hops to the destination. Note that the DFD may also be composed in order to optimize other parameters like battery power or end-to-end delay. The other potential forwarders (e.g., node A) overhear this further relaying and cancel their scheduled transmissions of the same packet. The rebroadcast packet is also received by the previous transmitting node and acknowledges the successful reception at another node. Simultaneously, the neighbors of the rebroadcasting nodes also received the packet and they determine if they are within the forwarding area relative to node B and destination D. Potential forwarders calculate an *Add_Delay* and compete to rebroadcast the packet again.

If no node is located within the forwarding area, greedy routing fails. This is detected if a node does not overhear a further rebroadcast within $Max_Delay + \epsilon$ of its previously broadcasted packet. This node forwards the packet further in backup mode. Therefore, the node broadcasts a request for a beacon packet. All neighbors that receive this packet reply with a beacon indicating their positions. The packet is then forwarded to the replying node that is closest to the destination. If none of the neighbors is closer to the destination than the requesting node, the packet is routed according to the face routing algorithm based on the "right-hand" rule, a concept known for traversing mazes, on the faces of a locally extracted planar subgraph, see for example GOAFR [7] for more details. As soon as the packet arrives at a node closer to the destination than where it entered backup mode, the packet switches back to greedy mode.

2.2 Simulations

We implemented and evaluated BLR in the Qualnet network simulator [8]. The results are given with a 95% confidence interval. Radio propagation is modeled with the isotropic two-ray ground reflection model. The transmission power and receiver sensitivity are set corresponding to a nominal transmission range of

250m. We use IEEE 802.11b on the physical and MAC layer operating at a rate of 2 Mbps. The simulations last for 900s. The simulation area is 6000m x 1200m and nodes move according to the random waypoint mobility model. The minimal and maximal speeds are set to 10% of an average speed of $20\,m/s$ for simulating a highly dynamic network. We consider speed as a proxy for any kind of topology changes, caused by either mobility, sleep cycles, interferences, adjustment of transmission and reception parameters, etc. The parameter Max_Delay is set to $2\,ms$. We compare BLR with a standard position-based routing protocol, namely GFG/GPSR[9][10].



(a) Delivery ratio (b) Delay

Fig. 2. Comparison of BLR and GFG/GPSR

In Fig. 2, the delivery ratio and the end-to-end delay are shown for different network densities. For low-density networks, the delivery ratio of BLR and GFG/GPSR are almost equal because packets are routed frequently in backup mode. The backup mode of BLR is similar to face routing of GFG/GPRS, except for the fact that it is reactive. The low delivery ratio of both protocols is due to temporarily partition of the network. For denser networks, the delivery ratio increases for BLR to almost 100% whereas GFG/GPSR is not able to deliver more than 90%. BLR outperforms GFG/GPSR especially in terms of end-to-end delay. The delay remains unaffected by the node density and below 30ms. GFG/GPSR on the other hand has a delay of at least 200ms, which is even increasing for higher node densities. Extensive evaluations revealed that the reasons for the much longer delays of GFG/GPSR are mainly threefold. First, nodes broadcast beacons periodically and for a dense network this may congest the network. Secondly, for higher node densities the chosen next hop is closer to the transmission range boundary and has a higher probability of not being available anymore and having left the transmission range. And third, due to the high mobility, packets loop between nodes as the stored position about neighbors does not correspond to the actual physical location of that node. For

a more comprehensive description of the protocol and additional simulation and analytical results cf. to [11].

3 Dynamic Delayed Broadcasting Protocol (DDB)

3.1 Protocol Description

We assume again that nodes are aware of their own position through any kind of mechanism. The only information required by DDB in order to broadcast a packet throughout the network is that each node knows its position and the position of the last broadcasting node as given in the packet header. DDB achieves local optimal broadcasting by applying the principle of Dynamic Forwarding Delay (DFD) which delays the transmissions dynamically and in a completely distributed way at the receiving nodes ensuring nodes with a higher probability to reach new nodes transmit first. Nodes that receive the broadcasted packet use the DFD concept to schedule the rebroadcasting and do not forward the packet immediately. From the position of the last visited node stored in the packet header and the node's current position, a node can calculate the estimated maximal additional area that it would cover with its transmission.

The explicit DFD function is crucial to the performance of DDB and should fulfill certain requirements in order to operate efficiently. The function should yield larger delays for smaller additional coverage and vice versa. In this way, nodes that have a higher probability to reach additional nodes broadcast the packet first. For simplicity reasons, we assume the unit disk graph as the network model and thus a transmission range scaled to 1. Taking into account the maximal additional covered area $AC_{MAX} \simeq 1.91$, which is achieved when a node B is located just at the boundary of the transmission range of node A, we propose a DFD which is exponential in the size of additional covered area, as it was shown in [12], that exponentially distributed random timers can reduce the number of responses. Let AC denote the size of the additionally covered area, i.e. $AC \in [0, 1.91]$,

$$Add_Delay = Max_Delay \cdot \sqrt{\frac{e - e^{\left(\frac{AC}{1.91}\right)}}{e - 1}} \tag{2}$$

where Max_Delay is the maximum delay a packet can experience at each node. A node does not rebroadcast a packet if the estimated additional area it can cover with its transmission is less than a *rebroadcasting threshold* which also may be zero. The objective of (2) is to minimize the number of transmissions and at the same time to improve the reliability of the packet delivery to all nodes. Like for BLR, it might also be optimized again for other parameters like battery power or network lifetime.

If a node receives another copy of the same packet and did not yet transmit its scheduled packet, i.e., the calculated DFD timer did not yet expire, the node recalculates the additional coverage of its transmission considering the previously

received transmissions. From the remaining additional area, the DFD is recalculated which is reduced by the time the node already delayed the packet, i.e., the time between the reception of the first and the second packet. For the reception of any additional copy of the packet, the DFD is recalculated likewise. Obviously, DDB can "only" take locally optimal rebroadcasting decisions as nodes receive only transmissions from their immediate one-hop neighbors and thus have no knowledge of other more distant nodes which possibly already partially cover the same area.

3.2 Analytical Evaluation

We want to calculate the expected size of the additional area AC that is covered by a node's transmission when using (2) as delay function.

Let $k \leq n$ denote the k-most distant neighbor of the sending node, i.e., $k = n$ and $k = 1$ yield the most distant and the closest neighbor respectively. Obviously, the k-most distant neighbor has also the k-largest additionally covered area. The expected value E_{AC}^{nk} for the additional coverage of the k-most distant neighbor is then solely depending on the number of neighbors n, cf. [11] for a detailed derivation.

$$E_{AC}^{nk} = \frac{2\Gamma(n+1)\Gamma(k+\frac{1}{2})}{\Gamma(k)\Gamma(n+\frac{3}{2})} \tag{3}$$

We compare this result with the expected additional coverage E_{AC}^{*} of other broadcasting protocols where the sequence of neighbors' transmission is independent of their additional coverage. Then the expected additional coverage is independent of the number of neighbors n and the same for all neighbors and therefore is constant.

$$E_{AC}^{*} = \frac{4}{3}$$

In Fig. 3, the graph is plotted for E_{AC}^{nk} of DDB and E_{AC}^{*} of other broadcasting algorithms depending on the number of neighbors $n = 1 \ldots 30$. Again, $k \leq n$ denotes the k-most distant neighbor. E_{AC}^{*} is simply the plane at $\frac{4}{3}$. Already for very few neighbors, the "best" node, i.e., $k = n$, already covers almost the maximum size of additional area. Furthermore, the next $k \leq n$-best nodes cover normally more than $\frac{4}{3}$ what would be covered by a node's transmission with other stateless broadcasting schemes. We can conclude that we might expect an improved performance up to $43\% = \frac{1.91}{4/3}$ in terms of numbers of transmissions. However, the advantage of DDB is not only the reduction in number of transmissions, but also that the delay can be reduced as distant nodes which transmit first add almost no delay.

As it is difficult to assess the exact influence of the MAC layer and to take into account the dependencies between neighboring nodes when their transmission ranges overlap, this analysis only provide a rough kind of boundary for the performance. For a more comprehensive description of the protocol and additional simulation and analytical results again cf. to [11].

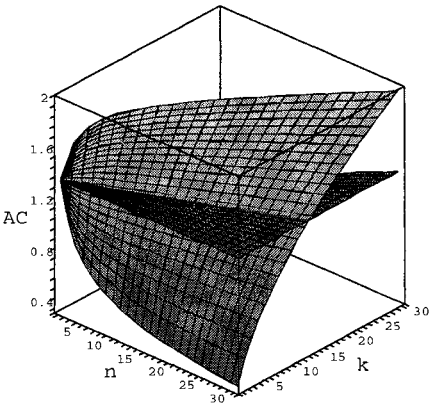


Fig. 3. Expected additional coverage

4 Conclusion

In this paper, we presented the Beacon-Less Routing Protocol (BLR) and the Dynamic Delayed Broadcasting Protocol (DDB) which are both based on a new paradigm where forwarding decision are no longer taken at a sender of a packet but in a completely distributed manner at the receivers. The paradigm is enabled by the broadcast property of the wireless medium such that packets are always simply broadcasted to all neighbors instead of addressing them to one specific neighbor. We implemented this new paradigm by a concept of Dynamic Forwarding Delay (DFD), where each receiving node dynamically delays the forwarding of received packets solely based on information given in the received packets and information available at this node itself. This new paradigm has three main advantages in ad-hoc networks.

- The fact that nodes do not require knowledge about their neighborhood allows reducing control traffic such as the broadcasting of beacons which can be reduced almost completely. This in turn conserves scarce network resources such as battery power and bandwidth and especially in dense networks allows the protocols to operate efficiently as no congestion occurs due to control traffic.
- In this paper we used a metric for DFD to reduce the number of transmission and hops as always the most "distant" node forwarded the packet first. The concept of DFD however can also easily be adapted to support optimizations for other metrics such as battery level, network lifetime, end-to-end delay.
- The protocols are almost completely stateless as no information on the network topology is used and, thus, they are unaffected by even very high rates of topology changes and prove highly scalable in terms of number of

nodes. The delay can be reduced up to an order of magnitude compared to other position-based protocols where neighbor positions are very inaccurate in highly dynamic networks.

References

1. M. Mauve, J. Widmer, H. Hartenstein, A survey on position-based routing in mobile ad-hoc networks, IEEE Network 15 (6) (2001) 30–39.
2. B. Williams, T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, in: Proceedings of the 3rd ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '02), Lausanne, Switzerland, 2002, pp. 194–2002.
3. E. M. Royer, C.-K. Toh, A review of current routing protocols for ad-hoc mobile wireless networks, IEEE Personal Communications Magazine 6 (2).
4. T. Camp, Location information services in mobile ad hoc networks, Tech. Rep. MCS-03-15, The Colorado School of Mines, Golden, CO, USA (Oct. 2003).
5. M. Heissenbüttel, T. Braun, T. Bernoulli, M. Wälchli, BLR: Beacon-less routing algorithm for mobile ad-hoc networks, Elsevier's Computer Communications Journal (Special Issue) 27 (11) (2004) 1076–1086.
6. M. Heissenbüttel, T. Braun, M. Wälchli, T. Bernoulli, Optimized stateless broadcasting in wireless multi-hop networks, in: IEEE Infocom 2006, Barcelona, Spain, 2006.
7. F. Kuhn, R. Wattenhofer, A. Zollinger, Worst-case optimal and average-case efficient geometric ad-hoc routing, in: Proceedings of the 4th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '03), Annapolis, Maryland, USA, 2003, pp. 267 – 278.
8. Qualnet (Nov. 2006).
URL http://www.qualnet.com/
9. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, in: Proceedings of the 3th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '99), Seattle, USA, 1999, pp. 48 – 55.
10. B. Karp, H. T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in: Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '00), Boston, USA, 2000, pp. 243–254.
11. M. Heissenbüttel, Routing and Broadcasting in Ad-Hoc Networks, Ph.D. thesis, University of Bern, CH-3012 Bern, Switzerland (Jun. 2005).
12. J. Nonnenmacher, E. W. Biersack, Scalable feedback for large groups, IEEE/ACM Transactions on Networking 7 (3) (1999) 375–386.

Benutzerorientierte Leistungs- und Verfügbarkeitsbewertung von Internetdiensten am Beispiel des Portals hamburg.de

Martin Gaitzsch

Universität Hamburg
Fakultät für Mathematik, Informatik und Naturwissenschaften
Department Informatik - Arbeitsgruppe TKRN
Vogt-Kölln-Straße 30, D-22527 Hamburg

Zusammenfassung. Der vorliegende Beitrag gibt einen Überblick über die Inhalte der Diplomarbeit *„Benutzerorientierte Leistungs- und Verfügbarkeitsbewertung von Internetdiensten am Beispiel des Portals hamburg.de“*, die in der Arbeitsgruppe TKRN unter Betreuung von Prof. Dr. Bernd E. Wolfinger entstand. Die Arbeit zeigt auf, wie durch eine Kombination von System-, Last-, Leistungs- und Verfügbarkeitsmessungen sowie durch die Vermessung adäquat gewählter Teilsysteme in sehr gezielter Weise nicht nur Engpässe in komplexen Internetkonfigurationen aufgedeckt und präzise eingegrenzt werden können, sondern auch, welche Optimierungsentscheidungen daraus betreiberseitig abgeleitet werden können.

1 Motivation

Das Internet wächst seit seiner Entstehung mit großer Geschwindigkeit und ein Ende des Wachstums ist nicht absehbar.

Mit der Größe des Internets ist auch seine Bedeutung gewachsen. Das *World-Wide-Web* mit allen darauf aufbauenden Diensten (Online-Shopping, Online-Auktionen, Online-Banking, Ticketreservierungen, ...) und Kommunikation per E-Mail sind für viele Menschen nicht mehr wegzudenken. Für Unternehmen ist ein Internetauftritt zur Darstellung eigener Produkte und Dienstleistungen heutzutage unverzichtbar geworden. Einige Unternehmen erwirtschaften ihren Umsatz ausschließlich mit Internetdiensten. Das Internet hat somit für Unternehmen eine noch größere Bedeutung als für Privatpersonen.

Viele Unternehmen, die für ihre im Internet angebotenen Dienste eine Verfügbarkeit rund um die Uhr benötigen, übersehen Schwachstellen im Netzdesign und der Konfiguration der Server. Dabei entsteht für ein Unternehmen durch den Ausfall seines Internet-Servers neben dem reinen Imageschaden unter Umständen auch konkreter wirtschaftlicher Schaden, der nach [1] bis zum Konkurs des Unternehmens führen kann. Dabei ist es nicht ausreichend, die Server vor Ausfällen zu schützen. Vielmehr müssen auch weitere Faktoren wie die Kapazitätsplanung oder die Anbindung an das Internet zum Erreichen hoher Verfügbarkeit berücksichtigt werden.

Für den Erfolg eines Internetangebots ist aber nicht nur die Verfügbarkeit entscheidend. Vielfach wird der Aspekt der Leistungsfähigkeit (Performance) übersehen; kein Anwender akzeptiert auf Dauer lange Wartezeiten [2]. Unternehmen, die Kunden langfristig an sich binden wollen, sind gut beraten, neben der ständigen Verfügbarkeit auch die Performance ihrer Dienste zu gewährleisten. Genauso wie für hohe Verfügbarkeit müssen auch zum Erreichen hoher Leistung verschiedene Faktoren berücksichtigt werden. Schwachstellen können u.a. die Rechenleistung der Server, die Anbindung der Server an das Internet oder der Internetanschluss eines Benutzers sein. Aufgrund des schnellen Wachstums des Internets ist auch die Prognose der Leistung bei steigenden Benutzerzahlen von Bedeutung, damit entsprechende Erweiterungsmaßnahmen rechtzeitig geplant und durchgeführt werden können.

2 Zielsetzung

In der Diplomarbeit wird die Leistungs- und Verfügbarkeitsbewertung von Internetdiensten am Beispiel des Portals „hamburg.de“ durchgeführt. Es werden unter verschiedenen Gesichtspunkten Mess- und Bewertungsmethoden erörtert und an den Internetservern des Portals angewendet. Die „hamburg.de GmbH & Co. KG“ ist ein Unternehmen, welches seinen Umsatz ausschließlich mit Internetdiensten erzielt. Die Ergebnisse haben somit auch für die Praxis eine hohe Relevanz.

Einsatzgebiete einer Systembewertung sind u.a. der Vergleich existierender Systeme , die Entwicklung neuer Systeme und die Analyse realisierter Systeme. Im Fall hamburg.de steht die Analyse und Optimierung der Leistung des realisierten Systems im Vordergrund. Die Leistung aus Benutzersicht wird u.a. durch den Internetanschluss eines Benutzers, die Anbindung der Server an das Internet sowie durch die Rechenleistung der Server bestimmt. Ein Ziel ist deshalb die Untersuchung der Teilbereiche zur Identifikation von Engpässen.

Leistungs- und Verfügbarkeitsanalysen können mittels Messungen oder Modellierung durchgeführt werden. Aufgrund der Gelegenheit, ein existierendes System zu untersuchen, stützen sich die Analysen der Diplomarbeit auf Messungen. Nach [3] dienen Messungen zum einen zur Untersuchung des Leistungsverhaltens realer Systeme, zum anderen zur Gewinnung realitätsnaher Einflussgrößen für die Modellierung. Messungen können entweder aus Benutzer- oder aus Betreibersicht durchgeführt werden.

3 Stand der Forschung

Zahlreiche Forschungsarbeiten untersuchen die vielfältigen Aspekte des Internets und der darauf aufbauenden Dienste. Ein großer Teil dieser Arbeiten befasst sich mit Messungen oder verwendet Messungen als Grundlage für Modellierungsstudien. Einen guten Überblick über Messmethoden und -werkzeuge vermittelt [4]. Eine sehr ausführliche Darstellung ist in [5] zu finden.

[6] stellt Methoden und Werkzeuge zur Analyse von Messdaten aus Verkehrsmessungen im Internet vor. Mit der Zusammensetzung von Internetverkehr beschäftigt sich unter anderem [7]. In dieser und weiteren Arbeiten wurde übereinstimmend ein Anteil von über 90% TCP-Verkehr am gesamten IP-Verkehr gemessen. In [7] wurde weiterhin ein Anteil von 35% HTTP-Verkehr am gesamten TCP-Verkehr ermittelt, was die große Bedeutung von Webdiensten verdeutlicht.

Mit Messungen und Messmethoden von verschiedenen Kenndaten von Internetpfaden befasst sich neben weiteren [8]. [9] untersucht darüber hinaus die Stabilität von Internetrouten. Mit Methoden zur Erstellung von Internet-Topologien beschäftigt sich unter anderem [10]. Lastcharakterisierung und -modellierung von Webservern ist Gegenstand der Arbeit [11] und weiteren. [12] behandelt die Modellierung und Transformation von Internetlasten. Zur Leistungsverbesserung von Webdiensten wird in [13] ein neues Anwendungsprotokoll vorgeschlagen, welches im Gegensatz zu HTTP nicht ausschließlich auf TCP sondern zusätzlich auf UDP basiert.

4 Aufbau der Arbeit

Für die Leistungs- und Verfügbarkeitsbewertung von Internetdiensten bietet sich eine Dekomposition des Gesamtsystems in zwei Komponenten an. Abbildung 1 stellt die Aufteilung dar, der die Gliederung der Arbeit folgt.

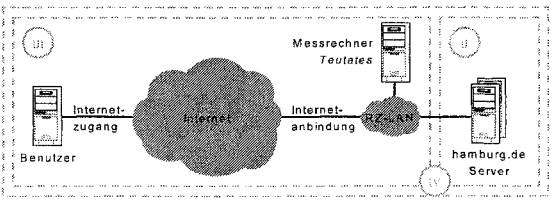


Abb. 1. Aufbau der Arbeit: Server- und Lastcharakteristiken (Teil II), Kommunikationsnetz (Teil III), Gesamtsystem (Teil IV)

In Kapitel 3 und 4 (Teil II) werden Server- und Lastcharakteristiken unabhängig vom Kommunikationsnetz untersucht. Kapitel 5 und 6 (Teil III) betrachten Eigenschaften des Kommunikationsnetzes zwischen den Benutzern und Servern von hamburg.de. Da die Untersuchungen ohne Einfluss der Server durchgeführt werden sollten, wurde für diese Arbeit ein dedizierter Messrechner (namens „Teutates") im Rechenzentrum von hamburg.de installiert. Nach Betrachtung der Teilkomponenten werden in Kapitel 7 bis 10 (Teil IV) verschiedene Untersuchungen des Gesamtsystems durchgeführt. Auch hier wird der Rechner „Teutates" benötigt, beispielsweise zur Lastgenerierung. Die einzelnen Kapitel verfolgen jeweils zu Beginn eines Kapitels definierte Ziele und sind in sich abgeschlossen.

5 Ergebnisse der Arbeit

Vor den Untersuchungen legt Kapitel 2 die Grundlagen in den Bereichen Internetdienste sowie Leistung und Verfügbarkeit in Bezug auf Internetdienste und Lastmodellierung. Es wird die Funktionsweise der betrachteten Internetdienste vorgestellt und eine typische Realisierung am Beispiel von hamburg.de dargestellt. Für Leistung und Verfügbarkeit werden Definitionen und Bewertungsmöglichkeiten erörtert.

In Kapitel 3 wird die Beziehung zwischen Zugriffen auf Webdienste und CPU-Auslastung der Webserver untersucht. Dabei stellt sich heraus, dass die Unterscheidung von nur einer Auftragsklasse bei dem untersuchten *Vignette*-System für eine Korrelation von Zugriffszahlen und CPU-Auslastung nicht ausreicht, aber schon die Kombination der zwei Auftragsklassen „Zugriffe Webseiten" und „Zugriffe sonstige Objekte" unter Einsatz der Methode der kleinsten Quadrate zu sehr guten Ergebnissen führt. Der durchschnittliche relative Fehler zwischen prognostizierter und gemessener CPU-Auslastung betrug 2,43%. Zur Validierung der Ergebnisse wurde mit derselben Parametrisierung die CPU-Auslastung eines weiteren Tages prognostiziert. Hier lag der durchschnittliche relative Fehler bei 3,35%, sodass die Genauigkeit des Verfahrens als sehr gut bewertet werden kann. Eine weitere Erkenntnis ist, dass ein Zugriff auf eine Webseite im Mittel 37× mehr CPU-Auslastung erzeugt als ein Zugriff auf ein sonstiges Objekt. Somit ist beispielsweise das Einfügen weiterer Bilder auf eine Webseite für die Auslastung der Webserver kaum von Bedeutung. Durch die Kenntnis, welche Zugriffsklassen die CPU-Auslastung in welchem Maße bestimmen, können Prognosen für eine steigende Anzahl von Zugriffen erstellt werden und Dienstangebote in Bezug auf ihren Ressourcenbedarf optimiert werden.

Kapitel 4 analysiert Zugriffe auf Web- und Maildienste in Hinsicht auf Benutzermodellierung. Es werden Benutzerverhaltensautomaten [14] erstellt, die den Kommunikationsablauf und das Verhalten der Benutzer abbilden. Die Auswertung von realen Protokolldateien der hamburg.de-Benutzer ermöglicht die realistische Parametrisierung der Modelle. Bei der Untersuchung der Maildienste ergab sich das überraschende Ergebnis, dass mit 40% ein sehr großer Anteil von Aufträgen an die Mailserver durch die Überwachungsmechanismen der vorgeschalteten „Loadbalancer" verursacht wird. Die durch Benutzer erfolgten Zugriffe wurden mit einem Benutzerverhaltensautomaten modelliert. Es zeigte sich, dass die meisten Benutzer automatisiert nach festgelegten Zeitintervallen E-Mails abrufen und dass einige Benutzer dabei extrem kurze Zeitintervalle wählen, was bei über 90% aller erfolgreichen Anmeldungen zur Übertragung von 0 Byte Nutzdaten führt. Es wird gezeigt, dass auch eine deutlich geringere Frequenz der Loadbalancer bei der Überprüfung der Funktionsfähigkeit der einzelnen Mailserver nur eine geringe Auswirkung auf die Verfügbarkeit hätte. Der entwickelte Benutzerverhaltensautomat kann unter anderem als Grundlage für Modellierungsstudien und für die Erzeugung realistischer synthetischer Hintergrundlasten genutzt werden. Die Untersuchung der Webdienste ergab einen großen Anteil an Zugriffen auf ein nicht existierendes Objekt, die durch Varianten des Wurmes „Bagle" erzeugt wurden. Zur Unterscheidung von Benutzern standen

ausschließlich die IP-Adressen zur Verfügung. Durch Plausibiltätsbedingungen wie die Anforderung von mindestens einer Webseite und zehn Bildern wurden „gültige" Benutzer definiert. Es stellt sich heraus, dass die vollständige Parametrisierung eines nicht-trivialen Benutzerverhaltensautomaten an den verfügbaren Informationen der Protokolldateien scheitert. Die Auswertung der dennoch zu gewinnenden Informationen ergab, dass die meisten Benutzer nur eine oder wenige Webseiten abrufen und die Besuchsdauer kurz ist.

In Kapitel 5 werden Eigenschaften von Internetzugängen der Dienstbenutzer untersucht. Dazu werden Kenndaten wie Verzögerung und maximale Datenrate repräsentativer Internetzugänge in Bezug auf die Server von hamburg.de gemessen. Reale und künstliche Netzhintergrundlast simulieren die Auswirkungen paralleler Benutzeraktivität und künstlich verringerter Datenrate auf die Antwortzeiten beim Abruf von Webseiten. Bei keiner der Messungen gab es Paketverluste und die Datenraten aller Zugänge wurden durch a priori bekannte Engpässe limitiert, was auf einen sehr guten Ausbau des benutzten Teils des Internets schließen lässt. Antwortzeitmessungen beim Abruf von Webseiten unter gleichzeitiger Netzlast zeigten, dass beim sequentiellen Abruf von Webseiten und zugehörigen Objekten die Download-Datenrate des verwendeten DSL-Anschlusses nur zu ca. 60% ausgenutzt wird. Der Grund dafür liegt im Slow-Start-Algorithmus von TCP. Aus den Ergebnissen folgt, dass auch eine noch höhere nominelle Datenrate keine merkliche Verbesserung der Antwortzeiten bewirken würde.

Im Gegensatz zu Kapitel 5 steht in Kapitel 6 die Anbindung der diensterbringenden Server an das Internet im Vordergrund. Es wird eine allgemeine Methode zur Erkennung von Engpässen vorgestellt und am Beispiel hamburg.de angewendet. Dazu werden Messungen von 15 verschiedenen ISDN-Providern durchgeführt und ein Netzstrukturplan aufgestellt. Dieser zeigte, dass hamburg.de an drei verschiedene Netze angebunden ist. Die ermittelten Datenraten unterschieden sich bei allen verwendeten Providern um weniger als 1% und bei den Messungen der RTT (Round-Trip-Time) gab es bei 15.000 versendeten Paketen keine Paketverluste. Die Schwankungen der RTT waren bei allen Providern sehr gering und die Ergebnisse der Tag- und Nachtmessungen unterschieden sich nur geringfügig. Somit waren keine Engpässe in der Anbindung der Server von hamburg.de an das Internet erkennbar. Nur die absolute Höhe der RTT differierte zwischen den Providern um bis zu einem Faktor von drei. Die in Kapitel 7 durchgeführten Messungen zeigten aber, dass dies beim Abruf von Webseiten nur geringe Auswirkungen hat.

In Kapitel 7 werden Leistungsmessungen von Webdiensten mittels Benutzersimulation durchgeführt und ein Ansatz zur gezielten Leistungsverbesserung betrachtet. Dazu wurden gleichzeitig Messungen von typischen Internetzugängen (vgl. Kapitel 5) und dem Messrechner „Teutates" direkt im hamburg.de-Rechenzentrum durchgeführt. Durch den Vergleich der netzabhängigen Antwortzeiten mit den netzunabhängigen Referenzergebnissen vom Rechner „Teutates" kann für jede Kombination aus Dienst und Internetanschluss ermittelt werden, welcher Anteil der Gesamtverzögerung durch den Server und welcher durch die Datenübertragung verursacht wird. Das Vorgehen erscheint als sinnvoller Ansatz zur

Optimierung von Webdiensten, weil die sehr stark variierenden Ergebnisse von mehr als 90% Serveranteil bis zu mehr als 99% Netzanteil zeigen, dass unbedacht eingesetzte Maßnahmen nahezu wirkungslos bleiben können. Mit den Ergebnissen ist hingegen eine gezielte Optimierung möglich. Je nachdem, in welchem Subsystem der Hauptteil der Verzögerung entsteht, werden zahlreiche Maßnahmen zur Leistungsverbesserung vorgeschlagen und bewertet.

In Kapitel 8 wird untersucht, wie sich die Last von Webservern auf die Leistung aus Benutzersicht auswirkt. Dazu werden die Webserver mit Anfragen belastet und die Last schrittweise bis zur vollständigen Auslastung erhöht. Gleichzeitige Messungen von verschiedenen Internetanschlüssen ermitteln die Antwortzeiten aus Benutzersicht. Durch die Auswertung von Protokolldateien wurde eine realitätsnahe Grundlast definiert und die Server mit vielfachen dieser belastet. Gleichzeitig von einem DSL- und einem ISDN-Zugang aus durchgeführte Antwortzeitmessungen beim Abruf von Webseiten zeigten, dass die absolute Ladezeitverlängerung beim ISDN- und DSL-Zugang in allen Lastzuständen nahezu identisch und somit vom Internetzugang unabhängig ist. In Bezug auf die CPU-Auslastung der Webserver steigen die Antwortzeiten zunächst mäßig an und die Zahl der Ausreißer erhöht sich. Bei sehr hoher CPU-Auslastung steigen die Antwortzeiten stark an. Die relative Erhöhung der Antwortzeiten ist vom DSL-Zugang aus deutlich höher als vom ISDN-Zugang. Bei zunehmender Verbreitung von schnellen Internetzugängen wachsen demnach die Anforderungen an die Serverleistung, da die Server nicht mehr weitgehend folgenlos im Bereich hoher Last betrieben werden können. Mit den Ergebnissen kennt der Betreiber die Charakteristiken seiner Systeme und kann besser planen, wann bei steigender Last Systeme aufgerüstet oder ausgetauscht werden müssen, um den Benutzern eine bestimmte Leistung zu bieten.

Eine Möglichkeit für Leistungsmessungen aus Benutzersicht ist die Simulation von Benutzern, wie sie beispielsweise in Kapitel 7 durchgeführt wird. Kapitel 9 untersucht hingegen, wie die von den *realen* Benutzern wahrgenommene Leistung ermittelt werden kann. Die Simulation von Benutzern bietet den Vorteil vollständige Kontrolle über die Randbedingungen von Experimenten zu haben und damit Reproduzierbarkeit zu gewährleisten. Möchte man aus den Experimenten Rückschlüsse auf alle Benutzer ziehen, ergeben sich Probleme. Zum Beispiel kann die Verteilung von Internetanschlüssen der Benutzer nur durch Schätzung ermittelt werden. Bei Untersuchung der realen Benutzerleistung liegt der Vorteil in der Berücksichtigung aller beeinflussenden Faktoren. Es werden u.a. die Auswirkungen von Clientsoftware und -hardware, gleichzeitiger Netzlast und unterschiedlichen Internetanbindungen implizit mitberücksichtigt. Wenn die Messungen der realen Benutzerleistung hinreichend präzise durchgeführt werden können, ergibt sich für den Betreiber ein sehr gutes Bild über die von allen Benutzern wahrgenommene Leistung seiner Dienstangebote. Für die Messungen wurden verschiedene Verfahren vorgeschlagen und bewertet. Dabei zeigte sich, dass die Leistungsmessung realer Benutzer keine triviale Aufgabe ist und alle vorgeschlagenen Verfahren Vor- und Nachteile haben. In der Arbeit wurden mit den „Zwei-Zählpixel-Verfahren“ innovative Methoden zur Leistungsmessung realer

Benutzer entwickelt und auf der Startseite des Portals hamburg.de exemplarisch implementiert. Die Verfahren beruhen auf geschickter Kombination der Techniken JavaScript, Cookies und serverseitiger Skriptsprachen. Mittels Referenzmessungen mit verschiedenen Webbrowsern und Internetzugängen konnte gezeigt werden, dass die Genauigkeit sehr gut ist. Sie konnte unter Berücksichtigung der Ergebnisse der Referenzmessungen mit einer einfachen Ergebnistransformation sogar noch gesteigert werden. Die „Zwei-Zählpixel-Verfahren" bieten für die Leistungsmessung realer Benutzer von Webdiensten demnach sehr hohe Genauigkeit bei vergleichsweise einfacher Implementation und universeller Einsetzbarkeit.

In Kapitel 10 werden Langzeitmessungen zur Verfügbarkeitsbewertung der Web- und Maildienste von hamburg.de durchgeführt. Durch gleichzeitige Messungen direkt im Rechenzentrum und über vier verschiedene Provider wurden sowohl die Diensterbringung an sich, als auch die Erreichbarkeit der Server aus dem Internet überprüft. Die Bewertung erfolgte unter Berücksichtigung einer erweiterten Verfügbarkeitsdefinition. Dazu wurden zunächst fehlerfreie von fehlerhaften oder außerhalb der messbedingten Zeitschranke erfolgten Antworten unterschieden. Der Anteil an fehlerfreien Antworten innerhalb der messbedingten Zeitschranke war bei keiner Kombination von Dienst und Provider niedriger als 99,65% und wurde maßgeblich durch planmäßige Wartungsarbeiten beeinflusst. Wird weiterhin die maximale Wartebereitschaft eines Benutzers niedriger als die messbedingte Zeitschranke definiert, dann ergeben sich unter Umständen deutlich geringere Verfügbarkeiten. Die Darstellung der Ergebnisse wurde mit einer „modifizierten" Verteilungsfunktion so gewählt, dass die Zeitschranke der maximalen Wartebereitschaft im Nachhinein festgelegt werden kann.

6 Fazit und Ausblick

Insgesamt betrachtet konnte dank der vielfältigen Untersuchungen eine umfassende Leistungs- und Verfügbarkeitsbewertung der Internetdienste von hamburg.de durchgeführt werden.

In einer kapitelübergreifenden Gesamtbetrachtung sind vor allem die folgenden Ergebnisse der Arbeit hervorzuheben:

- Eine effiziente Leistungsverbesserung von Webdiensten ist erst nach der Identifikation von Engpässen durch den gezielten Einsatz von Optimierungsmaßnahmen möglich.
- Die Leistungsbewertung realer Benutzer von Webdiensten ist eine schwierige Aufgabe, für die in dieser Arbeit eine innovative Methode entwickelt wurde.
- Leistungsmessungen an realen Systemen ermöglichen die Kapazitätsplanung unter Berücksichtigung der Leistung aus Benutzersicht.
- Der Ausbau des Internets ist im untersuchten Teilbereich ausgezeichnet.

Für die Vorgehensweise bei der Erstellung dieser Arbeit stellten sich die folgenden Punkte als besonders wichtig heraus: Aufgrund der hohen Komplexität des Gesamtsystems ist eine isolierte Betrachtung abgeschlossener Teilbereiche unabdingbar. Zum anderen ist bei der Durchführung von Messungen neben einer

zielorientierten Vorgehensweise die präzise Definition der Messschnittstellen von entscheidender Bedeutung.

Die Diplomarbeit zeigt zahlreiche Ansätze für weitere Forschungsarbeiten auf, beispielsweise eine detaillierte Untersuchung des Verhaltens verschiedener Webbrowser und deren Konfigurationen (z.B. Anzahl der parallelen Verbindungen) in Bezug auf die Leistung aus Benutzersicht sowie die damit erzeugte Netz- und Serverlast und deren Auswirkungen. Des Weiteren könnten dank der zahlreichen in dieser Arbeit gewonnenen Messdaten nunmehr Modellierungsstudien mit deutlich reduziertem Aufwand durchgeführt werden.

7 Literatur

1. S. Casper, „How to design your Web site for availability", SunWorld Online Magazin, 1999
2. A. Bouch, N. Bhatti, A. J. Kuchinsky, „Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service", Proc. ACM Conf. on Human Factors in Computing Systems, April 2000, 297-304
3. M. Haas, W. Zorn, „Methodische Leistungsanalyse von Rechensystemen", Oldenbourg, 1995
4. C. Williamson, „Internet Traffic Measurement", IEEE Internet Computing, Vol. 5, No. 6, November/Dezember 2001, 70-74
5. V. Paxson, „Measurements and Analysis of End-to-End Internet Dynamics", Ph.D. Thesis, Computer Science Division, UCB, April 1997
6. C. Courcoubetis, V. A. Siris, „Procedures and tools for analysis of network traffic measurements", Performance Evaluation, 48(1-4), Mai 2002, 5-23
7. M. Arlitt, C. Williamson, „An Analysis of TCP Reset Behavior on the Internet", ACM/SIGCOMM Computer Communication Review, Vol. 35, No. 1, Januar 2005, 37-44
8. C. Dovrolis, P. Ramanathan, D. Moore, „Packet-Dispersion Techniques and a Capacity-Estimation Methodology", IEEE/ACM Transactions on Networking, Vol. 12, No. 6, Dezember 2004, 963-977
9. Y. Zhang, V. Paxson, S. Shenker, „The Stationarity of Internet Path Properties: Routing, Loss, and Throughput", ACIRI Technical Report, Mai 2000
10. N. G. Duffield, F. L. Presti, „Network Tomography From Measured End-to-End Delay Covariance", IEEE/ACM Transactions on Networking, Vol. 12, No. 6, Dezember 2004, 978-992
11. G. Bai, C. Williamson, „Workload Characterization in Web Caching Hierarchies", Proceedings of IEEE/ACM MASCOTS, Oktober 2002, 13-22
12. B. E. Wolfinger, M. Zaddach, K. D. Heidtmann, G. Bai, „Analytical modeling of primary and secondary load as induced by video applications using UDP/IP", Computer Communications 25, 2002, 1094-1102
13. M. Rabinowich, H. Wang, „DHTTP: An Efficent and Cache-Friendly Transfer Protocol for the Web", IEEE/ACM Transactions on Networking, Vol. 12, No. 6, Dezember 2004, 1007-1020
14. B. E. Wolfinger, „Characterization of Mixed Traffic Load in Service-Integrated Networks", Systems Science Journal, Vol. 25, No. 2, 1999, 65-86

A System for in-Network Anomaly Detection

Thomas Gamer

Institut für Telematik, Universität Karlsruhe (TH), Germany

Abstract. Today, the Internet is used by companies frequently since it simplifies daily work, speeds up communication, and saves money. But the more popular the Internet gets the more it suffers from various challenges like DDoS attacks. This work, therefore, proposes an anomaly-based system that is able to detect adverse events caused by such challenges. The detection of network anomalies, in contrary to signature-based systems, ensures that previously unknown adverse events can be detected, too. Furthermore, the proposed system is designed for deployment within the network to allow a detection of adverse events as fast as possible, i.e., not only at the victim's edge of the network. To achieve such an in-network anomaly detection the system is designed hierarchically and applies refinement of detection granularity.

1 Introduction

Today's networks are threatened by challenges that appear with increasing frequency and comprise various kinds of attacks as well as unintended network problems. Challenges currently threatening networks include attacks like denial-of-service (DDoS) attacks [1] and worm propagations [2]. Furthermore, unintended network problems due to misconfigured nodes or flash-crowd events [3] pose a threat to today's networks, too. An automatic detection of adverse events caused by such challenges is still a problem for network operators. Additionally, autonomic networking will be a topic in the near future. Such networks need a mechanism to detect adverse events and apply suitable countermeasures autonomously.

With DDoS attacks an attacker does not exploit a weakness of the victim's operating system or application but aims to overload resources like link capacity or memory by flooding the system with more traffic than it can process. The attack traffic is generated by many slave systems which the attacker has compromised before. The attacker only has to coordinate all these slave systems to start the attack nearly at the same time against a single victim. Internet worms on the other hand exploit security holes in operating systems or applications to infiltrate a system. Afterwards, they start to propagate themselves to as many other systems as possible. One side effect of this propagation is the increasing bandwidth consumption since more and more worm instances try to propagate themselves to other systems. Today's countermeasures to worms are signature-based detection systems scanning for well-known worms.

An early detection of adverse events caused by such challenges allows a fast reaction and, thus, ensures a suitable protection of the network, the victims, and the network's resources. This requires a detection system within the network. Programmable networks enable a router to flexibly set up new services on that router, i.e., within the network. Therefore, programmable networks are suitable to achieve an in-network deployment of a detection system for adverse events. Such a detection system, however, has to face some difficulties, too. One of those is the fact that – in the worst case – the detection takes place within high-speed networks. This means that, though an on-line analysis is performed by the detection system, a negative impact on that router's forwarding performance must be avoided. Therefore, we propose the usage of a hierarchical detection system that applies refinement, i.e., detection granularity and analysis effort are adapted to the current stage of the detection. Such a system, therefore, works resource-saving and ensures that – even if it is built completely in software – there is no affection of a router's forwarding performance.

Furthermore, we propose to use an anomaly-based detection since various challenges, e.g. DDoS attacks, cannot be detected by a signature-based system due to their usage of protocol-conform packets. An anomaly-based detection system, however, analyzes traffic behavior and, therefore, can detect such challenges as well as previously unknown adverse events. Lastly, a signature-based system is only applicable in high-speed environments if it uses special-purpose hardware since it has to inspect each packet deeply.

If a DDoS attack, for example, is running error messages are generated by routers close to the victim as soon as the victim is not reachable anymore. Such changes of the traffic can be detected by combining various anomalies. In case of worm propagations an anomaly-based detection system can collect hints on such an adverse event e.g. by analyzing the ratio of error messages due to closed ports to the total number of connection requests. Such error messages are generated by scanned systems that are not vulnerable to this specific worm.

This paper details on a system for in-network anomaly detection. It is organized as follows: section 2 presents a short introduction to packet selection mechanisms. Section 3 details on the main characteristics of the detection system – a hierarchical architecture and refinement. Furthermore, architecture details are given for an example scenario. An evaluation of this example scenario then is described in section 4 and finally, section 5 gives a short summary.

1.1 Related Work

There are some existing approaches for DDoS attack detection that use special-purpose hardware: [4] uses network processors to perform a deep packet inspection of all observed packets in a backbone network. [5] uses special-purpose hardware to add a timestamp to each packet and then, does the analysis off-line.

Other anomaly-based approaches either cannot be applied in high-speed networks due to their high resource consumption or perform only a very coarse-grained detection without further refinement. The pushback mechanism [6], for example, is activated as soon as congestion occurs on a router. In this case a

flooding attack is assumed and a rate limiter is installed for the highest bandwidth aggregate of dropped packets. This approach has several disadvantages: an attack can be detected not until congestion occurs on a router and hence a detection is only possible at the edge of the network. Furthermore, no further verification is done if the rate limited aggregates really belong to an attack. Sterne et al. [7] detects stochastic anomalies by using a threshold-based DDoS detection mechanism on active networking nodes but no further refinement is done if an attack has been detected. Bro [8] is an open source network intrusion detection system that applies refinement. But – unlike our approach – the refinement has a different scope. Bro is an event-driven system and consists of three parts: the packet capture, the policy-neutral event engine, and the policy layer. A problem of this approach is that Bro creates lots of state by deep packet inspection and semantic analysis. Finally, the MVP architecture of Cisco Systems [9] uses refinement for detection of DDoS attacks, too, but this refinement is not very flexible and is only done in two steps, i.e., multiple stages are not possible for refinement.

2 Packet Selection

A packet selection mechanism is used especially in high-speed environments to reduce the number of packets that have to be inspected by a specific application, e.g. measurement or intrusion detection. The IETF working group PSAMP [10] proposed two types of packet selectors: filtering and sampling. *Filtering* is used if only a particular subset of packets is of interest. Filtering schemes are always deterministic and are based on packet content or router state. Therefore, filtering schemes are not suitable for a detection of adverse events. Any attacker who knows the filtering rules can adapt his challenge in a way that his packets are not selected by the detection system. This makes bypassing of the detection system easy. In contrast to filtering, *sampling* is used to infer knowledge about an observed packet stream without inspecting all packets. Therefore, only a representative subset of packets is selected which enables an estimation of properties of the total traffic. Sampling methods are either nondeterministic or do not depend on packet content or router state. The sampling methods are further grouped into two categories: random sampling and systematic sampling.

Systematic count based sampling is an example for a systematic sampling method. This sampling method is deterministic but independent of packet content and router state. For this method a *sampling interval* is defined consisting of a *selection interval* and a *non-selection interval*. A periodic trigger defines the beginning of a sampling interval. The unit of the intervals is count based. An example of this sampling method with a sampling interval of 5 packets, a selection interval of 2 packets, and a non-selection interval of 3 packets is shown in figure 1.

A sampling mechanism effectively reduces the number of packets that are inspected but it also introduces estimation errors. Thus, the parameters of the

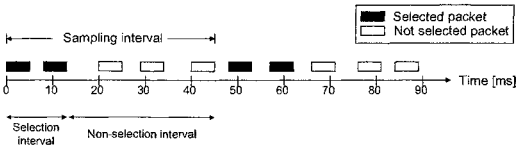


Fig. 1. Example of packet selection with systematic count based sampling

applied sampling mechanism have to be chosen in such a way that the error caused by packet selection is restricted to a predefined tolerance level.

3 Architecture

The detection system is designed hierarchical, anomaly-based, and flexible. Due to the fact that an anomaly-based detection is performed – i.e. only traffic behavior is analyzed – a packet selection mechanism as described in section 2 can additionally be applied.

The hierarchical characteristic of the system allows to split the detection of adverse events into different stages: A basic stage that scans for *stochastic anomalies* is running all the time. Specialized stages are loaded on demand for a more detailed detection of adverse events. Thus, refinement of detection granularity is applied by the detection system, i.e., detection granularity is increased with each subsequently loaded detection stage (see fig. 2). The basic stage of the hierarchical detection system dedicates only low analysis effort – this stage does only a simple packet classification – in order to perform a coarse grained detection that scans for indications of an adverse event. Further stages then are loaded whenever an adverse event is assumed in the basic stage. These further stages analyze only a part of the whole packet stream due to the information about the assumed adverse event gathered by the basic stage. Therefore, the further stages are able to do a more fine grained detection by applying deeper packet inspection on the reduced packet stream. Thus, the detection system gathers more detailed information about the adverse event in each of the further stages by using a higher analysis effort. In this paper, the notion *packet stream* designates a link's total aggregated traffic whereas a set of packets with same characteristics, e.g., all TCP packets, is referred to as an *aggregate*.

In summary, the hierarchical architecture of the detection system and the application of refinement save resources by running a basic stage with low resource consumption all the time and by loading further stages not until a stochastic anomaly is detected in the basic stage.

In order to detect stochastic anomalies the basic stage divides the packet stream on the fly into intervals with a fixed length. Furthermore, aggregates of interest are defined for observation, for example all TCP or all UDP packets. Then, for each predefined aggregate the number of packets that belong to this aggregate is counted in every interval. To make the system self-adaptable to net-

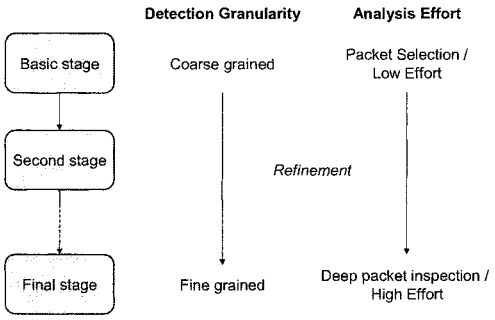


Fig. 2. Architecture of a hierarchical detection system using refinement

work load changes a dynamic *packet threshold* representing the average packet count in this aggregate for the last couple of intervals is calculated. At the end of each interval, for any aggregate a check is performed if the observed number of packets exceeds the packet threshold. To prevent the system from generating false positive indications and starting further stages for deeper inspections unnecessarily an *interval threshold* is defined. This interval threshold is necessary due to the self-similarity of internet traffic [11] which can cause normal traffic to exceed the packet threshold even though no adverse event is currently going on. Therefore, an indication only is generated if the packet threshold is exceeded in more consecutive intervals than the interval threshold. In addition to the detection of stochastic anomalies in the basic stage, the suspicious packet stream is scanned for further anomalies in specialized stages.

Flexibility of the detection system is ensured by usage of programmable networks. Since service modules are not tightly coupled to the packet forwarding but are loaded on demand, it is easy to update such modules or to add new service modules without a change to the rest of the system. Furthermore, the hierarchical architecture of the system allows the addition of new specialized stages. All characteristics described so far provide a flexible system for in-network anomaly detection that can be deployed in different environments like high-speed networks or small provider networks.

3.1 Small provider network

This section illustrates an exemplary architecture of the system for anomaly detection in case of a small provider network. In such a network detection of DDoS attacks and worm propagations is focused since these attacks are the most prevalent challenges. Therefore, the system scans for stochastic anomalies in the basic stage as described above. After detecting such a stochastic anomaly refinement is applied by loading two specialized consecutive stages (see fig. 3). The second stage uses a *distribution anomaly* to make a differentiation between DDoS attacks and worm propagations. This can be achieved by analyzing the distribution of packets into subnet prefixes based on destination addresses. Therefore,

the whole address space is divided into subnet prefixes based on the routing table of the node deploying the detection system. If large parts of the suspicious traffic – the number of packets by which the packet threshold was exceeded in the basic stage – are sent into exactly one subnet a DDoS attack is indicated since only one victim is currently attacked. If the suspicious traffic is equally distributed to all existing subnet prefixes a worm propagation is assumed since worms spread all over the internet. Based on the result of the second stage attack type specific protocol anomalies are scanned for in the third stage to identify either DDoS attacks or worm propagations in more detail. Currently, the anomalies used in our system for network anomaly detection offer no possibility to differentiate between DDoS attacks and legitimate traffic with the same characteristics, e.g. flash-crowd events [3].

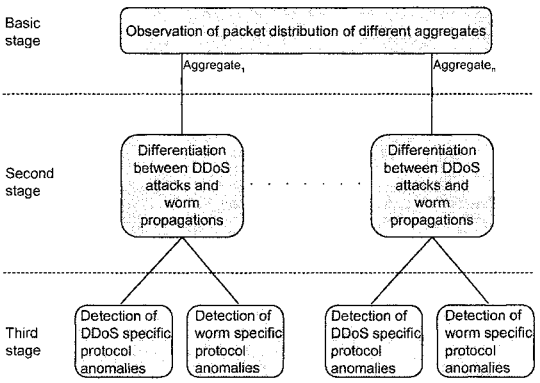


Fig. 3. Architecture of the detection system in a small provider network

In case of DDoS attacks the third stage of the detection system is mainly based on the fact that most of the existing DDoS attacks lead to a breach of symmetry between incoming and outgoing sub-aggregates which belong together by protocol definition. A TCP SYN flooding attack, for example, tries to exhaust a victim's open connection storage space by flooding the victim with TCP packets with SYN flag set. Due to the mass of connection requests the victim can only respond to a part of all requests by sending TCP packets with SYN and ACK flag set. All remaining requests are dropped and the victim sends no response at all if storage space is already exhausted. This leads to an asymmetry between incoming TCP packets with SYN flag set and outgoing TCP packets with SYN and ACK flag set which can be used to detect this kind of DDoS attack.

4 Evaluation

A prototype of the proposed detection system was implemented on a programmable platform. The basic stage of the detection system is the only service module

loaded at system startup. If this stage detects a stochastic anomaly in any aggregate, specialized service modules for further stages are loaded dynamically.

A network trace of real traffic with an average data rate of about 3 Mbit/s was used as background traffic of a simulation. Additionally, self-generated traffic was used representing a TCP SYN flooding attack with a packet rate of about 15 k packets per interval which corresponds to a data rate of about 0.8 Mbit/s. The average TCP traffic within the background traffic was about 1.7 Mbit/s. Due to the rather low bandwidth of background and attack traffic this evaluation is only a first step towards a small provider scenario but nevertheless, it shows that the mechanisms of the detection system work.

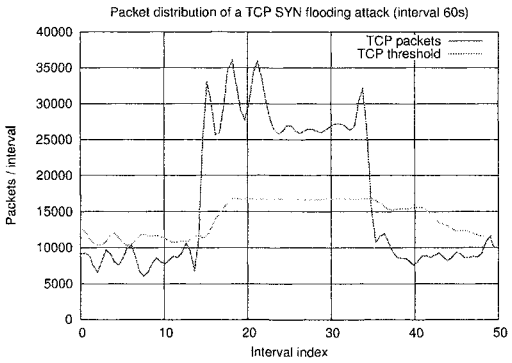


Fig. 4. Packet distribution of a TCP SYN flooding attack

The aggregated traffic – background and attack traffic – was analyzed by our detection system (see fig. 4). The red line shows the observed number of TCP packets per interval whereas the green line shows the packet threshold of the aggregate *TCP packets*. If an indication is generated by the basic stage, the threshold remains constant while the attack is running. We can clearly see that the simulated attack begins in interval 14. The exceeding of the packet threshold in more consecutive intervals than the interval threshold in the TCP aggregate results in loading further stages of the detection system in interval 18. Then, refinement is applied , i.e., the second stage analyzes only the suspicious TCP aggregate in more detail. It scans for a distribution anomaly which provides a differentiation between a DDoS attack and a worm propagation. In the simulation one specific subnet prefix could be detected which most of the traffic is sent to. Thus, the third stage is loaded that again applies refinement and analyzes only those packets of the suspicious aggregate for DDoS-specific protocol anomalies that are sent into the suspicious subnet prefix. In our simulation the third stage was able to detect an asymmetry between incoming TCP SYN packets and outgoing TCP SYN-ACK packets as described in section 3.1. Thus, the system correctly detected the TCP SYN flooding attack.

5 Conclusion and Outlook

In this paper a system for in-network anomaly detection is presented which is hierarchical and applies refinement of detection granularity. Therefore, the system is able to detect various adverse events in different environments, e.g. in small provider networks by scanning for stochastic anomalies, distribution anomalies, and protocol anomalies. A simulation of a TCP SYN flooding attack shows that our anomaly-based system is able to detect DDoS attacks.

In this paper, the evaluation was done only with low-bandwidth background traffic. Thus, future research has to address evaluations using background traffic with a higher bandwidth to simulate a more realistic small provider network. Furthermore, some work has to be done to achieve a differentiation between challenges like DDoS attacks and legitimate traffic with similar characteristics.

6 Acknowledgements

I would like to thank the supervisors of my diploma theses – Dr. Marcus Schöller, Dr. Roland Bless, and Prof. Dr. Martina Zitterbart – for very valuable discussions and their important feedback.

7 References

1. A. Hussain, J. Heidemann, and C. Papadopoulos. A framework for classifying denial of service attacks-extended. Technical Report, USC, 2003.
2. C. Shannon and D. Moore. The spread of the witty worm. IEEE Security and Privacy, 2(4):46 – 50, 2004.
3. I. Ari, B. Hong, E. L. Miller, S. A. Brandt and D. E. Long, "Managing Flash Crowds on the Internet", Proc. 11th IEEE/ACM Int. Symposium on MASCOTS, 2003.
4. L. Ruf, A. Wagner, K. Farkas, and B. Plattner. A detection and filter system for use against large-scale ddos attacks in the internet backbone. Proc. 6th Annual International Working Conference on Active Networking (IWAN), 2004.
5. D. Sass and S. Junghans, I2MP – An architecture for hardware supported high-precision traffic measurement, Proc. 13th GI/ITG Conference MMB. 2006.
6. J. Ioannidis and S. M. Bellovin. Implementing pushback: Router-based defense against DDoS attacks. Proc. of NDSS Symposium, 2002. The Internet Society.
7. D. Sterne, K. Djahandari, R. Balupari, W. L. Cholter, B. Babson, B. Wilson, P. Narasimhan, and A. Purtell. Active network based ddos defense. dance, 00:193, 2002.
8. Vern Paxson, Bro: A System for Detecting Networks Intruders in Real- Time, Computer Networks, 31 (23 – 24), 1999.
9. Cisco Systems, Defeating DDoS attacks, White Paper. 2005
10. N. G. Duffield. A framework for packet selection and reporting. Internet Draft, Work in Progress, Internet Engineering Task Force, January 2005.
11. K. Park and W.Willinger. Self-similar network traffic: An overview. In Self-Similar Network Traffic and Performance Evaluation. Wiley Interscience, 1999.

Simulation-Based Evaluation of Routing Protocols for Vehicular Ad Hoc Networks

Sven Lahde

Technische Universität Braunschweig, Institut für Betriebssysteme und Rechnerverbund, Mühlenpfordtstraße 23, 38106 Braunschweig
lahde@ibr.cs.tu-bs.de

Abstract. Information exchange between vehicles on the road is an important issue for future automotive applications. In vehicular ad hoc networks (VANETs), vehicles can be interconnected without the need of additional infrastructure along the roadside. Since these networks are highly dynamic, routing of data packets is a challenging task. In order to analyze and compare the performance of routing protocols, we developed realistic mobility models for typical traffic scenarios. These were used for a simulation-based evaluation of four routing protocols: AODV, DSR, FSR and TORA. AODV and FSR showed promising results in our simulations, while TORA is completely inapplicable for VANETs.

1 Motivation

Exchanging information between moving vehicles on the road without the need of any stationary infrastructure has become an intensive field of research during the last years. Approaches like the one developed in FleetNet [1] tackle this task with the help of multi-hop ad hoc networks (MANETs). Thus, each vehicle that is equipped with the necessary communication hardware represents a network node that acts as wireless station and mobile router at the same time. This way, distant vehicles can communicate with the help of intermediate vehicles that are used for packet forwarding. The application range of such communications systems covers safety-related applications that may inform drivers about the end of a traffic jam, accidents or other danger spots in front of the road as well as applications that are intended to increase the driving comfort. Moreover, vehicles may connect to the Internet via gateways along the roadside as well.

The communication characteristics in Vehicular Ad hoc Networks (VANETs) are quite challenging since vehicles move at high speeds and thus the network topology as well as the communication conditions may vary heavily over time. The movement of vehicles is influenced by several factors like the type of the road, traffic density, weather, daytime, or even the driver's mood. Routing protocols have to cope with these heterogeneous conditions and need to adapt to frequent link changes continuously. Previous work on the evaluation of routing protocols for MANETs is often based on quite small network topologies, where nodes move at low speeds using simple mobility models. However, the mobility model

itself clearly affects the results of network simulations [2]. Thus, more complex vehicular mobility models are needed in order to obtain more reliable simulation results for evaluating the performance of routing protocols in vehicular scenarios.

The remaining part of the paper is organized as follows: In section 2 the evaluated routing protocols are presented. The vehicular mobility models that have been developed are introduced in section 3. Afterwards, the evaluation results are discussed in section 4. Finally, the paper is concluded in section 5.

2 Routing Protocols

Various routing protocols have been proposed for MANETs. All of these protocols are based on different metrics (e.g. number of hops, delay, or power consumption) and routing approaches (e.g. proactive vs. reactive). Moreover, several protocols are specially designed for certain application scenarios. None of these approaches is known to be well-suited for all scenarios. Their performance depends on a specific combination of delays for initial route discovery, routing overhead as well as the nodes' movements. In addition, communication patterns may also have effects on the protocols' efficiency.

Two main classes of ad hoc routing protocols can be distinguished: location-based and topology-based protocols. Location-based routing protocols forward data packets with the help of information about the geographic position of sender and destination (e.g. by using GPS). An indispensable requirement for using location-based protocols is the availability of location services and servers. For this reason, we focused on topology-based routing protocols that work without these additional services and determine routes using information on the network topology. For our evaluation we chose four protocols to be compared: Ad Hoc On Demand Distance Vector (AODV) [3] Dynamic Source Routing (DSR) [4], Fisheye State Routing (FSR) [5] and the Temporally-Ordered Routing Algorithm (TORA) [6]. AODV, DSR, and TORA belong to the class of reactive routing protocols that discover routes only when they are needed. These protocols suffer from a high initial packet delay since a route to a new destination has to be discovered first. Reactive protocols often use a response-reply mechanism for this, which differs in the way where routing information is stored and how packets are forwarded. In contrast, FSR discovers routes proactively. It maintains routes to all possible destinations in the MANET and updates the routing information continuously with the help of link-state messages that are exchanged between nodes. To reduce the protocol overhead, the update interval for distant nodes is higher than for nodes in the immediate vicinity.

3 Mobility Models

The use of specific mobility models has a significant effect on the results of network simulations. In previous work on the performance analysis of ad hoc routing protocols the simple Random Waypoint Model is often used. In this model, nodes randomly choose a new destination within a rectangular simulation

area and move towards it. Such simple models are completely inapplicable for modeling VANET scenarios. Hence, more realistic mobility models are needed that reflect the movements of vehicles on the road. However, since road traffic is influenced by a huge set of parameters, some simplifications are still needed. In our scenarios, all vehicles are assumed to be points moving on a lane. Moreover, we do not distinguish between different types of vehicles like e.g. passenger cars or trucks. In the following, a freeway and a city mobility model are introduced.

3.1 Freeway Mobility Model

For the Freeway Mobility Model, the simulation area is assumed to be a freeway section of a specific length that may have multiple lanes per direction. Vehicles on a freeway drive at high speeds and may accelerate very fast. Since freeways typically have several lanes per directions, vehicles may overtake other vehicles in order to drive at higher speeds. The vehicles' speed is determined by using the Intelligent-Driver-Model (IDM) [7] which is a macroscopic car-following traffic model. IDM is able to adapt a vehicle's speed to the speed of other vehicles driving ahead. In addition, it allows for a realistic modeling of a vehicle's approach to any obstacles in front. IDM is able to emulate vehicle movements that are close to reality. Moreover, it is based on a quite small set of parameters which can be evaluated on the basis of real traffic measurements.

A vehicle n's acceleration is determined according to 1 on the basis of the parameters shown in fig. 1. It depends on its current speed v_n, the net distance s_n to vehicle $n-1$ driving in front and the approaching rate $\Delta v_n = v_n - v_{n-1}$ to this vehicle. The first two terms in 1 represent the acceleration of a vehicle on an empty road based on its maximum acceleration a_n, the driver's desired speed v_{des} and the acceleration exponent δ. A possible brake retardation in the case that slower vehicles or obstacles are in front of the vehicle is described by the last term. It comprises the desired distance s^* to the obstacle or vehicle in front which is determined according to 2. s^* is affected by the minimal distance of two successive vehicles in a traffic jam ($s_{jam}^{(n)}$), the safe time headway T and the vehicle's comfortable deceleration b.

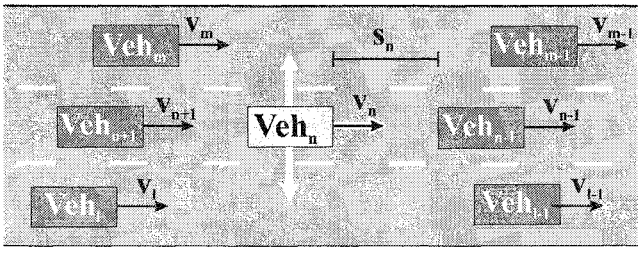


Fig. 1. Influencing variables on MOBIL

$$\dot{v}_n^{IDM}(v_n, s_n, \Delta v_n) = a_n \cdot \left[1 - \left(\frac{v_n}{v_{\text{des}}^{(n)}}\right)^{\delta} - \left(\frac{s^*(v_n, \Delta v_n)}{s_n}\right)^2\right] \quad (1)$$

$$s^*(v_n, \Delta v_n) = s_{\text{jam}}^{(n)} + T v_n + \frac{v_n \Delta v_n}{2 \cdot \sqrt{a_n b_n}} \quad (2)$$

In addition, a mechanism for modeling lane-changes is needed to emulate overtaking maneuvers of vehicles. The accurate modeling of lane-changes is very complex. Nevertheless, in our model we need an approach that is able to cope with a manageable set of parameters. Thus, the lane-chance strategy MOBIL (Minimizing Overall Breaking Induced by Lane-Changes) [7] has been used. MOBIL decides whether a vehicles should change its current lane with the help of the IDM accelerations of neighboring vehicles as shown in fig. 1. It determines, whether a lane-chance of vehicle n is advantageous for the local traffic situation. Therefore, it uses two criteria to come to a decision: the safety and the incentive criterion. The safety criterion ($a_{mn} \geq -b_{save}$) ensures that after the lane-change no accident can happen and other vehicles do not have to brake hard. If this criterion is met, the incentive criterion has to be evaluated before a vehicle is induced to change its lane. The incentive criterion analyzes the current and virtual IDM accelerations in the local traffic situation as illustrated in fig. 1. The criterion is given in 3 for a lane-change from the middle to the left lane. Besides the different IDM accelerations, it also includes a so-called politeness factor p that represents the willingness of a driver to change lanes although a faster vehicle comes from behind. Moreover, a threshold γ is used to avoid ping-pong effects. The incentive criterion is met, if the IDM acceleration of the examined vehicle and the weighted accelerations of neighboring vehicles after the lane-change are higher than the overall accelerations before plus the threshold γ.

$$a_{n(m-1)} + p(a_{mn} + a_{(n+1)(n-1)}) > a_{n(n-1)} + p(a_{(n+1)n} + a_{m(m-1)}) + \gamma \quad (3)$$

3.2 City Mobility Model

In urban areas vehicles drive at moderate speeds compared to the freeway scenario. The city road network is characterized by a large number of intersections, at which vehicles may have to stop and either turn off or drive straight on. Therefore, the traffic flow in cities can be characterized as very heterogeneous. The number of network reconfigurations can be expected to be very high. The road network in the City Mobility Model is modeled as a Manhattan-like grid with horizontally and vertically oriented street sections. All streets are assumed to have one lane per direction, although some main streets in larger cities may consist of several lanes per direction. Since overtaking maneuvers are atypical for urban traffic scenarios, we assume that vehicles are not allowed to overtake each other. Thus, MOBIL is not used in this scenario. At the start of a simulation, all vehicles are randomly distributed on the modeled street sections. A vehicle's

speed is determined with the help of the IDM as already mentioned above. Its approach to an intersection is modeled according to the approach to an obstacle. At each intersection, the vehicle has to reduce its speed until it stops in front of the intersection. In the next step, the vehicle chooses its new direction at a specific probability. A FIFO mechanism is used to control the traffic flow, if several vehicles reach the intersection at the same time.

Table 1 shows the whole set of constant parameters used. The values are based on related work about theoretical and practical evaluation of traffic flows.

Table 1. Constant parameters of IDM and MOBIL

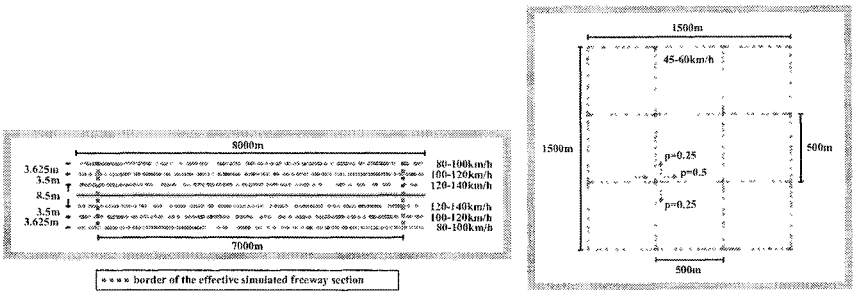
Param.	Description	Freeway	City
a	Maximum acceleration [m/s^2]	1.2	1.2
b	Comfortable deceleration [m/s^2]	1.5	1.5
δ	Acceleration exponent	4	3
s_{jam}	Minimum jam distance [m]	2	2
T	Safe time headway [s]	1.4	1.4
b_{save}	Limiting value of brake retardation [m/s^2]	0.5	
p	Politeness factor	$\in [0, 1]$	
γ	Lane-change threshold [m/s^2]	0.8	

4 Evaluation

The routing protocols are evaluated with the help of the network simulator ns-2 [8] in its version 2.27. This simulator is widely used for the evaluation of MANET protocols. We used the simulator's basic 802.11 MAC model and set the transmission range to 100 m. The simulation scenarios are shown in fig. 2.

In the freeway scenario, the simulated freeway section is assumed to have a length of 8 km. Since ns-2 requires that nodes reaching the end of the freeway section have to reappear at the beginning of another lane, we introduced guard distances of 500 m length at the beginning and end of the freeway section. The communication in these sections is ignored for the evaluation in order to avoid undesirable effects. The freeway's cross-section is modeled according to German regulations. A vehicle's desired speed is chosen between 80 km/h and 140 km/h depending on its initial lane. In the city scenario (fig. 2(b)) eight streets have been modeled each having a total length of 1500 m. The distance between parallel streets is 500 m. Since not all drivers keep the speed limit in urban areas, a vehicle's desired speed is chosen from an interval between 45 km/h and 60 km/h. At each intersection, a vehicle chooses a new direction at a specific probability shown in fig. 2(b). The values are based on UMTS reference scenarios.

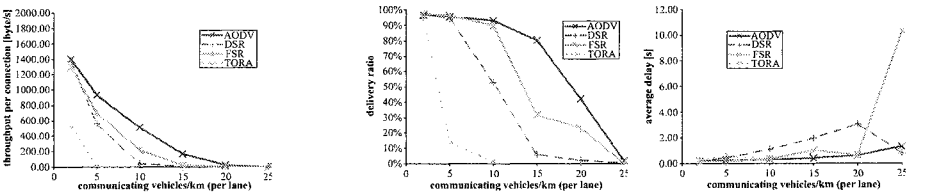
The network simulations have been performed at different traffic densities between 2 and 25 communication vehicles per km and lane. Thus, in the city



(a) Freeway (b) City

Fig. 2. Scenario characteristics

scenario 48 to 600 nodes have been simulated. In the freeway scenario the number of simulated node ranges between 96 and 1200 veh/km. The average number of TCP connections that are established between vehicles on the road is assumed to be half of the number of simulated nodes. Thus, on average each node participates in a connection. In the freeway scenario we also assumed that 80 % of the connections are established between vehicles driving in the same direction. For analyzing the routing protocols' performance, we chose common performance measures: end-to-end throughput, packet delivery ratio, and average end-to-end delay. Some of the simulation results are presented in the following.

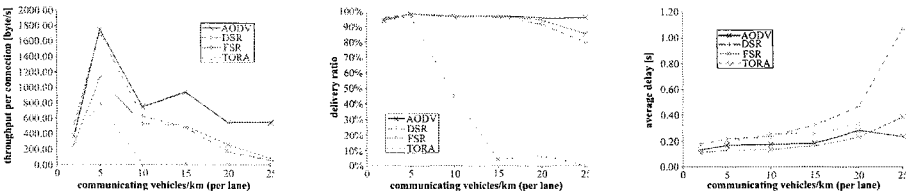


(a) Throughput (b) Delivey ratio (c) Delay

Fig. 3. Results of the Freeway Scenario

The results of the freeway simulations are shown in fig. 3. The TCP throughput per connection (subfig. (a)) shows an exponential decrease with increasing traffic density for all routing protocols resulting from the higher number of communicating nodes. AODV performed best in this scenario with a throughput of up to 1399.28 byte/s followed by FSR and DSR. While DSR's throughput at a traffic density of 2 veh/km was comparable to that of AODV, it decreases very fast at higher densities of 5 and 10 veh/km. Again, TORA performs worse compared to the other protocols. The delivery ratio shown in fig. 3(b) emphasizes these results. While the protocols delivered up to 94 % of the packets at a traffic density of 2 veh/km, only less than 2 % were delivered at the highest

simulated traffic density of 25 veh/km. The average end-to-end delay is depicted in fig. 3(c). At a low traffic density of 2 veh/km, FSR has the shortest delay (0.12 s), while AODV (0.16 s), DSR (0.18 s) and TORA (0.21 s) need longer to deliver data packets. The delays of AODV, and DSR increase up to a traffic density of 20 veh/km, where DSR has an average delay of 3.06 s. The results at a traffic density of 25 veh/km have to be take with a pinch of salt since only very few packets were delivered in this scenario. Thus, only few samples were available for determining the end-to-end delay. This fact explains the declining delay of DSR as well as the clear delay spike of FSR. Another reason for this delay spike is the huge amount of link state updates that were exchanged between nodes running FSR. This routing overhead consumes a large part of the available bandwidth.



(a) Throughput (b) Delivey ratio (c) Delay

Fig. 4. Results of the City Scenario

The results of the urban scenario are depicted in fig. 4. Subfig. (a) shows the average throughput per connection over the density of communicating vehicles. From a density of 2 veh/km to 5 veh/km, the throughput of all protocols clearly increases which results from the increased network connectivity (0.8 vs. 1.6 neighbors in a vehicle's transmission range). At higher traffic densities, the throughput declines again since more and more vehicles have to share a common wireless channel. All in all, AODV performs best in our simulations and outperforms the other protocols especially at higher traffic densities. TORA obviously has clear problems to deal with high traffic densities. The delivery ratio is shown in subfig. (b). The results underline the previous conclusions. Up to a traffic density of 5 veh/km all protocols are able to deliver between 93.56 % and 98.35 % of the data packets. From a density of 15 km/h on, the delivery ratio of FSR and DSR decreases in comparison to AODV. At a traffic density of 25 veh/km AODV still performs best (95.86 %) followed by FSR (85.22 %) and DSR (79.70 %). Finally, the average packet delay is presented in subfig. (c). The delays of all protocols range between 112.6 ms and 1.06 s. DSR has the highest delays at almost all simulated traffic densities, which relates to its route discovery mechanism. Up to a traffic density of 20 veh/km FSR was able to deliver packets fastest. All in all, it can be stated that AODV performs best in this scenario followed by FSR. DSR suffers from quite high delays, while TORA is completely inapplicable for VANETs in urban scenarios.

5 Conclusion

Information exchange between vehicles is a key requirement for future automotive applications to increase the driver's safety or comfort. However, routing data packets in such highly dynamic environments is still a challenging task. The common way to evaluate protocols in larger scenarios is doing network simulations since real-world experiments are complex and hardly reproducible. Important aspects for these simulations are realistic mobility models. In this paper we propose two mobility models that can be used to generate more realistic vehicular movement scenarios. These mobility models consider the adaptation of a vehicle's speed to other road users as well as overtaking maneuvers. With the help of these models, the performance of the routing protocols AODV, DSR, FSR and TORA has been evaluated. An important observation was that all routing protocols performed very heterogeneously. The results show that AODV runs best in the simulated scenarios. It achieves a very high throughput and was able to deliver packets quite fast. One reason for this is its low routing overhead. Especially in the city scenario, FSR was able to deliver packets fastest. In the freeway scenario it even achieved the second best throughput. However, the routing overhead of FSR increases clearly at higher traffic densities due to the link-state update messages. DSR suffers from high delays resulting from its source route mechanism. Finally, the simulation results indicate that TORA is completely unsuitable for VANET environments. Future work will include more complex road traffic scenarios as well as Internet gateways or communicating traffic signs along the roadside, which are integrated into the network.

References

1. Franz, W., Eberhardt, R., Luckenbach, T.: FleetNet - Internet on the Road, 8th World Congress on Intelligent Transportation Systems, Sydney, October 2001
2. Camp, T., Boleng, J., Davies V.: A Survey of Mobility Models for Ad Hoc Network Research, Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications **2** (2002) 5, 483–502
3. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing, IETF Request for Comments (RFC) 3561, July 2003
4. Johnson, D. B., Maltz, D. A., Hu, Y.-C.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), IETF Internet Draft, work in progress, draft-ietf-manet-dsr-09.txt, April 2003
5. Gerla, M., Hong, X., Pei, G.: Fisheye State Routing Protocol (FSR) for Ad Hoc Networks, IETF Internet Draft, work in progress,
6. Park, V., Corson, S.: Temporally-Ordered Routing Algorithm (TORA) Version 1 – Functional Specification, IETF Internet Draft, work in progress, draft-ietf-manet-tora-spec-04.txt, July 2001
7. Treiber, M., Helbing, D.: Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell, Contribution to the 16th Symposium "Simulationstechnik ASIM 2002" edited by Djamshid Tavangarian and Rolf Grützner, Rostock (Germany), September 2002, 514–520
8. The ns-2 Network Simulator, http://www.isi.edu/nsnam/ns/, July 2004

Dynamic Algorithms in Multi-user OFDM Wireless Cells*

James Gross

Telecommunication Networks Group
Technische Universität Berlin
Einsteinufer 25, 10587 Berlin, Germany
gross@tkn.tu-berlin.de

Abstract. This paper presents several results on dynamic OFDMA systems. It addresses especially the algorithmic complexity involved with several resource allocation approaches, sub-optimal heuristics for the use in practical systems, the related signaling overhead and modifications to the IEEE 802.11 protocol stack. It is argued that for the generation of dynamic OFDMA resource allocations very good sub-optimal methods exist while the loss due to signaling can be kept low for a large range of system parameters. Finally, an outline of the integration of dynamic OFDMA schemes into OFDM-based IEEE 802.11 systems is presented, providing significant performance benefits for such wireless local area networks.

1 Introduction

During the last ten years orthogonal frequency division multiplexing (OFDM) has become a very popular transmission scheme for frequency-selective broadband communication channels. Example systems that feature OFDM are digital video and audio broadcasting (DVB and DAB), wireless local area networks like IEEE 802.11a/g/n, wireless metropolitan area networks like IEEE 802.16, the upcoming extension of high-speed down-link packet access (HSDPA) in 3G cellular networks but also wired access systems like the digital subscriber line (DSL).

OFDM offers the advantage of mitigating intersymbol interference (ISI). By means of advanced signal processing the broadband channel is split into N narrowband sub-carriers (where N takes values between 52 and 2048). Each sub-carrier exhibits a frequency-flat behavior. Thus, instead of transmitting many digital symbols sequentially (as in a broadband single carrier system), N symbols are transmitted in parallel. Therefore, an OFDM system with an equivalent gross symbol rate can afford an N-times increased symbol time per sub-carrier, which reduces the impact of ISI significantly.

However, there is still frequency selectivity in the system as the channel gain varies over a larger set of sub-carriers. In addition, in a multi-user scenario, for example the down-link of a cell, the gain for each sub-carrier varies also regarding different terminals (referred to as multi-user diversity). This offered diversity can be exploited by

* This work has been supported partially by the German research funding agency 'Deutsche Forschungsgemeinschaft (DFG)' under the graduate program 'Graduiertenkolleg 621 (MAGSI/Berlin)'.

dynamic resource allocation schemes. Given the knowledge of sub-carrier gains at the transmitter, modulation/coding combinations as well as the transmit power can be allocated dynamically per sub-carrier (commonly referred to as bit- and/or power loading). Moreover, disjoint sub-carrier sets can be assigned to different terminals (known as dynamic sub-carrier assignments). These dynamic allocation schemes for multi-user scenarios, often referred to as dynamic OFDMA schemes, can improve various transmission metrics such as the total transmit power, error rates or system throughput. Despite this fact, many issues remain open regarding the application of such dynamic OFDM(A) schemes in practical systems.

In this paper three such aspects are addressed, summarizing the major contributions of [1]. Initially, the question arises how dynamic resource allocations should be performed in order to serve the data flows of several terminals best (i.e. maximizing the number of flows that can be served while maintaining fairness). Given this objective, an important issue for practical systems is how to generate such allocations in real time. As the sub-carrier gains are only stable for several milliseconds, potential allocation algorithms have to terminate quite fast. This problem is addressed in Section 3. Once resource allocations have been determined at some central point in the network, a further issue is how to convey necessary control information to the terminals (as they have to be informed of their next allocations). This so called signaling problem is discussed in Section 4. Finally, the integration of dynamic OFDMA schemes into OFDM based wireless local area networks (i.e. IEEE 802.11 a/g) is studied. Here, apart from the issue of computational complexity and the signaling overhead also backward compatibility and other protocol aspects play an important role (Section 5). Finally, some conclusions are drawn and future work is presented (Section 6).

2 System Model

Consider the down-link of a single wireless cell. The access point serves J terminals which all receive data flows (consisting of packets queued at the access point). A slotted system is assumed in which time is divided into units (frames) of duration T_f. A total bandwidth of B [Hz] at the center frequency f_c [Hz] is available for data transmission (maximum total transmit power of P_max). The given bandwidth is split into N OFDM sub-carriers, each one featuring a fixed symbol duration $\frac{N}{B} = T_\mathrm{s}$. Each sub-carrier gain varies due to path loss, shadowing and fading, hence the perceived signal quality (SNR) per sub-carrier varies from frame to frame (but is assumed to be constant during one frame). Depending on the SNR, M different amounts of bits might be represented per symbol for each sub-carrier (i.e. M different modulation/coding combinations are available). The choice of an adequate modulation/coding combination is determined by a constraint on the error probability.

Each frame is split into a down-link and an up-link phase. OFDMA is applied during the down-link phase. The duration of one down-link phase is denoted by T_d. A total of $S = T_\mathrm{d}/T_\mathrm{s}$ symbols per sub-carrier can be transmitted during that down-link period. Prior to each down-link phase, the access point generates new assignments of sub-carriers to terminals based on the knowledge of the sub-carrier states (i.e. the SNR). Perfect channel knowledge at the access point is assumed (by estimating the sub-carrier states during the previous up-link phase and assuming a reciprocal channel gain).

3 Dynamic Algorithms for Resource Allocation

Assume initially that the transmit power per sub-carrier is statically distributed. Then the gain per sub-carrier and terminal directly yields the SNR. Given the SNR, a certain modulation/coding combination is obtained, as the transmission is subject to a certain error constraint and simply the "best" modulation/coding combination (i.e the one with the highest throughput) is chosen which fulfills the error constraint. Denote by $b_{j,n}^{(t)}$ the amount of bits that can be transmitted to terminal j on sub-carrier n during down-link phase t. How should sub-carriers be assigned to terminals?

From a system point of view a basic constraint is that each sub-carrier should only be assigned once. Denote by the binary variable $x_{j,n}^{(t)}$ the decision if a sub-carrier/terminal pair is fixed as assignment during down-link phase t. Given this framework, a simple assignment strategy is to allocate sub-carriers per down-link phase such that the sum-rate of the cell is maximized [2]:

$$\max \quad \sum_{j,n} b_{j,n}^{(t)} \cdot x_{j,n}^{(t)} \qquad \text{s. t.} \quad \sum_{j} x_{j,n}^{(t)} \leq 1 \qquad \forall n \ . \tag{SUMRATE}$$

However, in a cell typically several terminals are located farer away from the access point than other stations. Hence, a subset of all terminals always has a much better SNR per sub-carrier than the remaining stations. As a consequence, terminals with a significantly lower sub-carrier gain obtain only occasionally a sub-carrier, which leads to starvation of their flows. In order to avoid this situation, a more sophisticated approach is required. Ideally, the assignment scheme should maximize the throughput of each terminal equally. This "max-min"formulation of sub-carrier assignments [3] is given by:

$$\begin{aligned} &\max \quad \varepsilon \\ &\text{s. t.} \quad \sum_{j} x_{j,n}^{(t)} \leq 1 \quad \forall n \qquad , \quad \alpha_j^{(t)} \cdot \sum_{n} b_{j,n}^{(t)} \cdot x_{j,n}^{(t)} \geq \varepsilon \quad \forall j \ . \end{aligned} \tag{MAXMIN}$$

Above $\alpha_j^{(t)}$ gives the opportunity to scale the minimum throughput per terminal j and down-link phase t. Hence, given the queue state of each terminal at the access point, the assigned rates for the next down-link phase can be scaled according to the queue sizes. Compared to a static approach (for example, a TDM approach where each terminal receives all sub-carriers for some down-link phase in a round-robin manner) it can be shown that a dynamic approach according to (MAXMIN) provides a much higher throughput per terminal [4]. However, the question remains open how the assignments can be generated in practical systems given the values $b_{j,n}^{(t)}$.

This question can be answered by considering the computational complexity. In fact it turns out that (MAXMIN) is *NP*-hard. PARTITION reduces in polynomial time to the decision problem of (MAXMIN) [1]. In addition, a broader set of dynamic OFDMA optimization problems all turn out to be *NP*-hard (obviously (MAXMIN) with dynamic power allocation, but also optimization problems where the transmit power is reduced for a given set of rates per terminal, referred to as margin-adaptive allocation problems [6]). Finally, practical instances of (MAXMIN) indeed turn out to be diffcult to

solve by software for linear integer programming problems (an example instance has been added to a database of difficult integer programming problems, cf. opt1217 of [5]). Thus, sub-optimal schemes are required for the application in real systems.

One particular good sub-optimal scheme is relaxation. By initially relaxing the integer constraints on the assignments $x_{j,n}^{(t)}$, (MAXMIN) becomes a pure linear programming (LP) problem. For LPs polynomial time algorithms are publicly available which perform quite promising even for bigger instances of (MAXMIN) (in the range of tens of milliseconds on standard computers without applying customization). Once the relaxed solution is obtained, a necessary step is to find a good feasible solution consisting only of integer assignments. For the case of static power distribution among the subcarriers, a simple rounding approach performs well (cf. Figure 1). In case that the power is distributed dynamically, a more advanced approach is required (cf. Figure 1). In summary, relaxation has been found to provide the best performance of all so far published schemes.

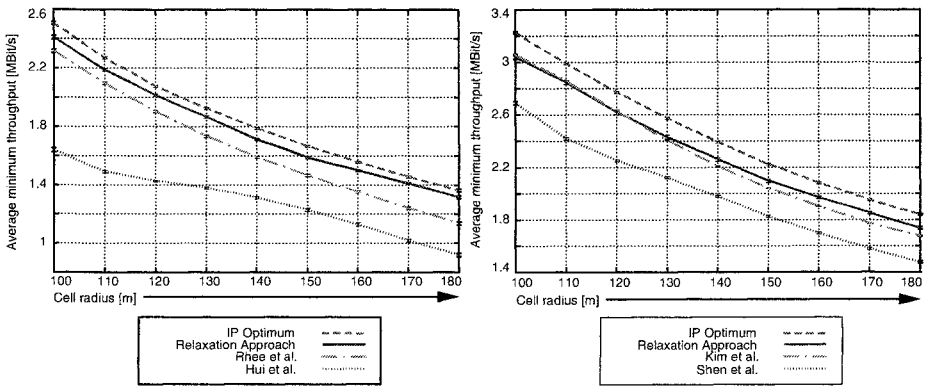


Fig. 1. Minimum throughput per cell for an IEEE 802.11a like wireless OFDM system with $J = 8$ terminals (for more information on the simulation scenario refer to [4]). Comparison of the IP optimum with the relaxation approach and several other sub-optimal schemes proposed in the literature. Left graph: static power distribution – Right graph: dynamic power distribution.

4 Signaling Overhead

Signaling is a problem which applies to many dynamic approaches in OFDM systems. It relates to the need to inform terminals about the next resource allocation such that the payload transmission can be successfully received. The access point basically has to inform each terminal which sub-carriers have been assigned to it during the next downlink phase. In addition, the terminals also have to be informed of the respective modulation/coding combination chosen per sub-carrier. A fundamental question regarding signaling is if the resulting overhead outweighs the benefits from dynamic OFDMA.

Initially, the following straightforward model is used to study the overhead. At the beginning of each down-link phase a signaling phase is inserted. The signaling information is broadcasted on all sub-carriers during this signaling phase using a predefined, fixed modulation type with bit rate b_{sig}. For each assignment the triple $\langle$sub-carrier, terminal, modulation$\rangle$ has to be transmitted. However, by transmitting all assignments per down-link phase, the sub-carrier reference can be omitted as the position of the tuple $\langle$terminal, modulation$\rangle$ indicates the sub-carrier it refers to. Therefore, a fixed number of $N \cdot (\lceil \log_2(J) \rceil + \lceil \log_2(M) \rceil)/(N \cdot b_{\text{sig}})$ signaling bits is required per down-link phase leading to $\varsigma = \lceil (\lceil \log_2(J) \rceil + \lceil \log_2(M) \rceil)/b_{\text{sig}} \rceil$ OFDM symbols of overhead. For payload communication $S - \varsigma$ symbols remain per down-link phase. This scheme is referred to as fixed-size signaling field model (FSSF).

From these basic considerations it is clear that the signaling overhead depends on various system parameters such as the number of sub-carriers N, the number of terminals J, the number of modulation/coding combinations M, but also on the length of a down-link phase T_{d} (as this determines the number of OFDM symbols S). Moreover, it turns out that there exists a trade-off between the control overhead and the achieved performance, as for example a larger number of sub-carriers leads to a higher average throughput per terminal (due to the increase of the symbol times but a fixed guard period setting) while also increasing the signaling loss. An example result of this is shown in the left graph of Figure 2. Clearly, the net throughput of the dynamic OFDMA scheme still outperforms the static approach. However, for an increasing number of sub-carriers the net average throughput per terminal first increases up to an optimum and sharply decreases thereafter. For a large number of sub-carriers the net throughput is even worse than for the static approach. Also note that considering a dynamic OFDMA system without signaling cost leads to very different system results. This is true for a broad set of system and environment parameters. The most important parameters for the overhead are the system bandwidth, the duration of a down-link phase and the number of sub-carriers.

This observed performance behavior motivates the investigation of the optimal net performance. In order to judge the efficiency of the FSSF model, a lower bound on the signaling overhead can been derived. The outcome of the assignment algorithm at the access point can be interpreted as an stochastic source of information (generating symbols from a discrete set). By exploiting the first- and second-order statistical properties of this information source, it is possible to derive the entropy (defined as the minimal binary rate required to represent the source without losses). This yields an upper bound on the net throughput. In the left graph of Figure 2 the upper bound on the net throughput from the entropy rate is shown as well, indicating that there is significant room for improvement. This performance improvement stems from correlation. Sub-carrier states are correlated in time and frequency. Hence, the stronger the correlation in time and frequency, the more likely is a correlated assignment of sub-carriers. The entropy indicates that for the considered scenario already a significant amount of correlation is present in the assignments. In order to exploit this correlation in practice, several options arise. First of all, the assignments could be better encoded by a more sophisticated representation. For example, only "new" assignments could be signaled, where "new" refers to the change of a sub-carrier assignment from down-link phase t to the next

one $t + 1$. A somewhat similar approach can be found for exploiting the correlation in frequency as well as exploiting the correlation in both dimensions. However, the downside of these schemes is that per single sub-carrier assignment more bits are required to represent them. Thus, if the correlation is low, the resulting overhead is high.

However, given these representations building on top of the correlation of assignments, a further option is to stimulate correlated assignments within the assignment algorithm itself. This can be driven to optimize the net throughput for a given representation. Assume that the bit cost per signaled assignment is given by C_{sig} (the more the representation exploits correlation, the higher is this cost). Then, for a given sub-carrier assignment (and possibly a given previous assignment), the total number of signaling symbols ς required to transmit the signaling information can be obtained. The net throughput depends on the amount of bits that can be transmitted per OFDM symbol during the payload phase but also on the remaining duration of the payload communication $S - \varsigma$. Ultimately, this leads to a quadratic, integer optimization problem for which an iterative algorithm has been developed [7]. This approach can achieve a significant performance increase for dynamic OFDMA systems as demonstrated in the right graph of Figure 2 (exploiting only the correlation in time and the correlation in time and frequency). Note that even with these optimization approaches a system can be subject to such rapid channel changes that the resulting signaling cost is too high. Then a static approach should be preferred.

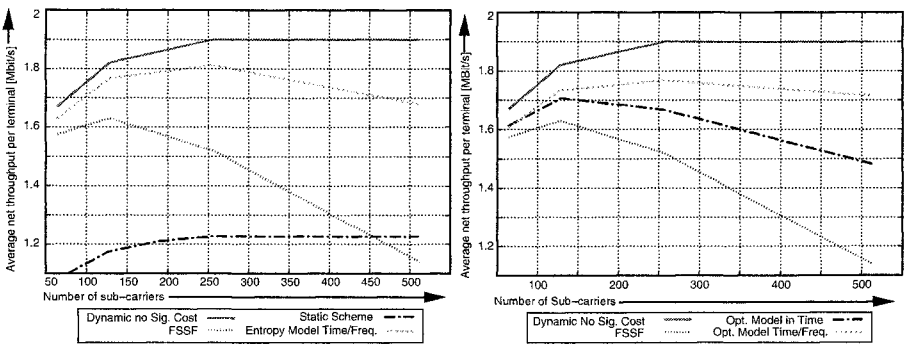


Fig. 2. The impact of signaling cost for different models (more information on the simulation scenario is given in [7]). Left graph: Basic comparison of the signaling cost modeled by the FSSF versus the cost derived from entropy analysis versus a static and dynamic OFDMA system without any signaling cost. Right graph: Two optimization approaches reducing the signaling overhead of the FSSF by exploiting the correlation in time and in time/frequency.

5 Integration into IEEE 802.11 a/g

Having discussed various aspects regarding dynamic resource allocation algorithms in OFDM systems, the question arises how such schemes could be incorporated in up-

coming or even existing OFDM based wireless networks. This will be discussed considering the IEEE 802.11 wireless local area network standard, focusing on the OFDM-based implementations (802.11a and 802.11g). For the infrastructure mode a dynamic OFDMA scheme fits best to the down-link transmissions from the access point to several stations. In the following the focus is only on this transmission direction.

Clearly, the OFDM-based IEEE 802.11 link layer and physical layer protocols have to be changed such that the access point can acquire the channel knowledge, afterwards calculate the sub-carrier assignments and finally transmit the payload (prepending a signaling field). In order to perform these tasks the transmission order is modified, as indicated in Figure 3. Initially, the access point transmits a Beacon frame as this allows it to access the medium with a higher priority. Immediately after the transmission of a Beacon a modified RTS frame is transmitted, which polls all stations to be included in the following OFDMA burst transmission. The stations reply with a legacy CTS frame which is used at the access point to estimate the sub-carrier states. Then the assignments are calculated and an OFDMA burst frame is generated which holds the signaling information. After the payload transmission the stations have to acknowledge the reception of their frames. Finally, the access point ends the transmission cycle by a RTS to itself.

Some issues come up when evaluating this modified layout. First of all, at the beginning of the transmission sequence the access point does not know the exact duration the medium will be busy, as the sub-carrier states are not known yet. Hence, the virtual allocation vector, referred to as NAV (network allocation vector), can not be set to the precise end of the transmission cycle. This potentially harms legacy devices. The problem can be resolved by initially (during the transmission of the Beacon frame at the beginning) setting the NAV to a very large value, blocking the channel for a long time. Once the precise length of the busy period is known (when the OFDMA burst frame is generated) the NAV is reset by all acknowledgement frames and by the final RTS frame. Hence, even the NAV of legacy devices can be controlled by this scheme. A further issue relates to the calculation time at the access point. Obviously, after acquiring the channel knowledge, the access point will have to calculate the assignments. However, the medium access scheme enables stations to access the wireless medium if it is not busy for a time span of DIFS (which is in the range of 30 μs for IEEE 802.11a/g). Hence, the calculation of the assignments and the generation of the OFDMA burst must not require more than this time span - otherwise some stations might interfere with the OFDMA burst transmission. This can be resolved by either transmitting a busy tone (which degrades the performance of the system, though) or by pipelining the assignment calculation. In this case the access point starts to compute the assignments for a subset of stations after acquiring their channel knowledge.

Regarding the performance, several issues have to be taken into account. Compared to a sequential transmission of the packets in the legacy mode, there is a higher overhead by the new scheme due to the initial Beacon frame, the signaling part and the trailing RTS to itself. In addition, for short packets in the legacy case no RTS/CTS frame exchange takes place, favoring the legacy system even more. However, only a single medium access takes place in this new proposal which gives it a significant performance improvement. In addition, during the payload transmission data is transmitted

with a much better efficiency. Finally, the new scheme controls the packet error rate as the modulation types are adapted per sub-carrier, leading to much less retransmissions especially compared to the legacy mode without RTS/CTS frame exchange. Initial results on this new mode show that it is very promising [1].

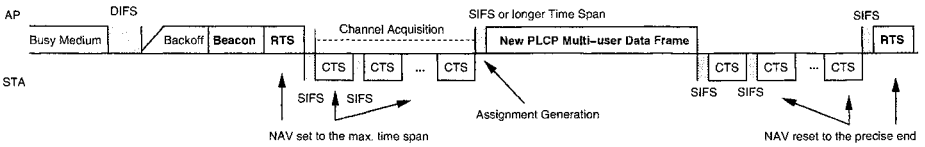


Fig. 3. New transmission sequence for a down-link data transmission in OFDM-based IEEE 802.11 systems featuring dynamic OFDMA.

6 Conclusions

This paper has summarized major results from [1] regarding dynamic OFDMA systems. Various aspects regarding the algorithmic complexity, the signaling overhead and the incorporation into existing protocol stacks have been addressed. It turns out that dynamic OFDMA systems provide a superior performance while the increase in complexity is much lower than at first hand assumed. Thus, dynamic OFDMA is a promising technique for future wireless systems.

References

[1] Gross, J.: Dynamic Algorithms in Multi-User OFDM Wireless Cells. PhD Thesis, Technische Universität Berlin (2006)

[2] Jang, J., Lee, K.: Transmit Power Adaption for Multiuser OFDM Systems. IEEE J. Select. Areas Commun. No. 2 (2003) 171–178

[3] Ergen, M., Coleri, S., Varaiya, P.: QoS Aware Adaptive Resource Allocation Techniques for Fair Scheduling in OFDMA Based Broadband Wireless Access Systems. IEEE Trans. Broadcast. No. 4 (2003) 362–370

[4] Bohge, M., Gross, J., Wolisz, A.: The Potential of Dynamic Power and Sub-Carrier Assignments in Multi-User OFDM-FDMA Cells. Proc. of IEEE Globecom, November 2005

[5] Koch, T.: MIPLIB. Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), http://miplib.zib.de/miplib2003.php, November 2006.

[6] Wong, C., Cheng, R., Letaief, K., Murch, R.: Multiuser OFDM with Adaptive Subcarrier, Bit and Power Allocation. IEEE J. Select. Areas Commun. No. 10 (1999) 1747-1758

[7] Gross, J., Geerdes. H., Karl, H., Wolisz, A.: Performance Analysis of Dynamic OFDMA Systems with Inband Signaling. IEEE J. Select. Areas Commun. No. 3 (2006) 427-436

Self-Organizing Infrastructures for Ambient Services

Klaus Herrmann

University of Stuttgart
Institute of Parallel and Distributed Systems (IPVS)
Universitätstr. 38, 70569 Stuttgart
klaus.herrmann@acm.org

Abstract The vision of Ambient Intelligence (AmI) as a new paradigm for supporting the mobile user in his daily activities is currently entering the focus of European research efforts. A high degree of autonomy on the part of the supporting software system is inherent to this vision of omnipresent and continuously running services. However, adequate concepts for creating respective infrastructures that may operate autonomously and in a self-organized fashion are still largely unexplored. We propose the *Ad hoc Service Grid* (ASG) as a dedicated AmI infrastructure that may be deployed in an ad hoc fashion at arbitrary medium-sized locations (shopping malls, construction sites, trade fairs, etc.). In this paper, we give an overview over our results thus far. We focus on the problems of service placement, discovery and lookup, and data consistency within an ASG environment and show how we have solved these problems with new self-organizing and adaptive algorithms. These vital functions are the basis for the realization of ASG systems and represent an essential contribution to AmI research in general.

1 Introduction

In recent years, the vision of *Ambient Intelligence* (AmI) [2] has produced considerable research efforts. The main idea of AmI is that mobile users may wirelessly access services (anywhere and anytime) that support them in their daily activities without putting any administrative burden on them. One essential building block of AmI are infrastructures that allow local access to local, facility-specific services. To enable services that enhance the user's interaction with his current environment, the physical surrounding of the user must be enriched with computing resources that enable service provisioning. One example scenario is a shopping mall that offers *ambient services* to customers, enabling them to navigate through the mall, find certain products quickly, and optimize the contents of their shopping cart, for example, according to the overall price or quality.

In this paper, we describe the *Ad hoc Service Grid* (ASG) infrastructure [5] that enables facility-specific ambient service provisioning at medium-sized locations (e.g., shopping malls, construction sites, trade fares, etc.). Our goal is to provide a service infrastructure that is easy to setup, flexibly scalable, and requires minimal administrative effort. The easy setup and the flexibility at the networking layer is achieved by adopting *Mobile Ad hoc Networking* (MANET) technology. To minimize the administration overhead, we propose a software layer that we call the ASG *Serviceware*. This software has the task of providing basic support for running services within an ASG in a self-organized fashion. That is, all aspects of executing services and adapting to changing

user demands shall be managed autonomously by the software. We will describe the core functions, algorithms, and protocols of the ASG Serviceware and show how they operate and interact in order to render the overall ASG self-organizing.

The rest of the paper is structured as follows. In Section 2, we discuss related work. The ASG model is introduced in Section 3 before we give an overview over the algorithms and protocols used to render its operation self-organizing in Section 4. In Section 5, we present our conclusions and give an outlook on future work.

2 Related Work

There are two major research strands in the area of AmI infrastructures at the moment. The first one deals with appropriate middleware systems. The research in the middleware area has been active for many years and is always quick in conquering new domains. Thus, a plethora of systems has been and continues to be proposed here. Cabri et al. propose the LAICA system as an agent-based middleware for AmI [1] and claim that agents "can naturally deal with dynamism, heterogeneity and unpredictability". Apart from that, no mechanisms are introduced that may substantiate this claim. The same is true for the SALSA system presented by Rodríguez et al. [11] for healthcare applications. Vallée et al. [14] seek to combine *multi-agent techniques* with *semantic web services* to create a system the can adapt to the user's current context in a context-aware service composition process. O'Hare et al. advocate the use of "agile agents" as a design principle for AmI [10]. These mobile agents are based on the *Beliefs Desires Intentions* (BDI) model that is well-known in intelligent agent research. However, no real motivation is given as to why this particular technology should be ideal for AmI.

The second strand originates from the area of *Service-Oriented Architectures* (SOA) and copes with the problems of finding, matching, and composing services in an AmI environment. Some variations of tools from the Web Services domain are being deployed for this purpose. Omnisphere is an architecture that supports the discovery of service components and the composition of higher-level services [12]. *Typed data flows* are used for service composition, and a matching mechanism is proposed that employs user preferences, devices capabilities, and the user's context to select the set of service components. Hellenschmidt et al. propose the SodaPop system that allows the composition of higher-level services from the components available on individual user devices [3]. They claim that their system enables devices to self-organize in order to collectively provide a service, but they do not substantiate this claim. Issarny et al. suggest to use a declarative language for specifying AmI systems [9]. The *Web Service Ambient Intelligence* WSAMI language allows the specification of composed services on the basis of Web Services technologies.

3 The Ad hoc Service Grid Model

3.1 Motivation

There are two obvious alternatives available for providing facility-specific, ambient services to mobile users. The first one is to use cellular phone networks as they are used

in a classical location-based service scenario. However, in this scenario, the available bandwidth is limited and any communication is expensive. The second alternative is the coverage with normal 802.11 (WLAN) access points which serve as a wireless extension of some wired infrastructure while services are being provided by some high-end server. This approach provides high bandwidths, and the communication is free of charge. However, it produces high costs for setting up the wired infrastructure. Moreover, it offers only limited flexibility and scalability since the wired infrastructure is fixed and cannot be easily changed.

We conclude that neither of the two technologies is particularly well-suited for the provisioning of facility-specific services. Therefore, we propose an alternative model that we explain in the following.

3.2 The Model

The ASG is based on the concept of *Service Cubes* (also called *Cubes* or *nodes* hereafter). A Service Cube is a PC-class computer that provides ad hoc networking capabilities and computing power. It has no peripheral devices (display or input devices), relies on a permanent power supply, and is equipped with a wireless network interface. In order to cover a given location with Ambient Services, a number of these Service Cubes is distributed over the location such that they can spontaneously set up a wireless network connecting all Cubes. Thanks to the concept of self-contained Service Cubes, an ASG network has a highly modular structure: New modules (Cubes) can be added and existing ones can be removed or repositioned to re-shape the network in an ad hoc fashion. This provides a flexible way of scaling an ASG to an appropriate size and allows for a quick and easy setup. No extensive planning and no construction work is necessary to cover a location in this way. It also enables different business models for providing an ASG since Service Cubes may be rented, collectively bought by different participants, or monolithically provided by some operator. Clients that wish to use ASG services may become a part of the ad hoc network by connecting to any of the Cubes with their mobile device. We assume the existence of some technology that allows for a seamless hand-over such that clients may move freely through the ASG network and always be connected to at least one Cube.

3.3 The Drop-and-Deploy Vision

The ultimate goal that we pursue with the ASG may best be characterized as *Drop-and-Deploy*: Anyone who decides to deploy an ASG simply has to distribute (drop) a certain number of Service Cubes at the respective location, switch them on, and *inject* the desired services at an arbitrary Cube. The structuring of the required software infrastructure, the binding of clients to services, and the adaptation to changing conditions is completely taken over by the ASG software. The road towards this goal holds some major challenges from diverse fields of computer science. In our work, we concentrate on a set of fundamental algorithms and protocols that are required in order to realize this vision of a self-organizing ASG infrastructure.

4 Algorithms and Protocols for a Self-Organizing ASG

Setting up a communication network is only the first step towards the operation of an ASG. In order to provide services within an ASG network, a software platform is required that can deal with the dynamics in the system. This *Serviceware* has to support the Drop-and-Deploy idea inherent to the ASG. In our work, we concentrate on three central aspects of such a Serviceware:

1. **Self-organizing service distribution:** Having a single replica of each service at some fixed Service Cube in an ASG may be sufficient to provide this service in a very basic way. However, if the ASG and the group of clients has a certain size, this solution is suboptimal. Temporary partitions in the network decrease the availability of the service. Requests have to be routed through the entire network, which wastes bandwidth, and increases response times. Therefore, it is vital to replicate services and to position them at specific Service Cubes such that most clients are served by a near-by replica. Moreover, this placement must not be static. It must be able to adapt if the client request patterns or the network topology change.
2. **Service discovery and lookup:** If services replicate and reposition dynamically, finding and using them becomes a challenge. Therefore, the idea of dynamic service placement directly implies the necessity for an adequate lookup service (LS) that is able to cope with this form of dynamics. This LS has to be distributed, too, and it has to apply some update strategy with an acceptably low overhead, such that updating the location information of repositioned replicas does not jam the network.
3. **Data consistency among stateful ASG services:** Most useful services that may be deployed in an ASG are stateful. That is, they store mutable data that may be read and written in a distributed way by clients. Therefore, an adequate consistency protocol is required that keeps the replicas of a service consistent with each other. This protocol, too, has to honor the specific conditions and the dynamics in an ASG.

In the following, we will examine each of these problems and our respective solutions in turn.

4.1 Distributed Service Placement

We have developed an adaptive service placement algorithm [6] that is run by each replica. This algorithm allows a replica to *migrate* from one node to another, to *replicate* (clone itself and subsequently migrate both clones), or to *dissolve* (remove itself from the system). The basic objective of this algorithm is to move replicas closer to the clients that use them. Additionally, the lookup service (cf. Section 4.2) enforces that a client always uses the service that is closest to it. These two principles define a feedback process: As a replica moves closer to its clients, it *attracts* even more clients from the respective area. This, in turn, increases the area's attractiveness for the replica which causes it to move closer, and so on. As the replica moves into the group of requesting clients, a negative feedback sets in and lets the replica converge to a stable position.

The placement algorithm itself is fully decentralized and requires no external control. It constantly inspects the incoming *message flows* and periodically takes a local

adaptation decision. This is possible without any additional communication between the replicas since they are coupled indirectly through the message flows they receive. This mechanism creates a stable, coordinated global placement that is adaptive to changes in the environment. The algorithm consists of three rules for the different adaptations:

1. **Idle Rule:** A replica dissolves (is removed) if it received less then α requests in a time interval m.
2. **Replication Rule:** A replica replicates to neighbor nodes u and w, if it receives a significant flow via u that consists of requests with an average path length of more than ρ hops. w is either set to the dominant node among the remaining neighbors, or the second replica stays at its current node. ρ is called the *replication radius*.
3. **Migration Rule:** A replica migrates to a neighbor node u, if the message flow that is coming in via u is stably *dominant* (larger than the sum of all remaining flows).

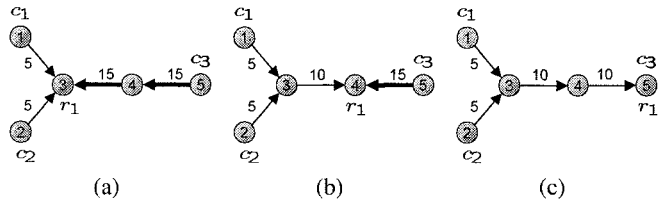


Figure 1. Exploiting dominating flows for incremental cost optimization.

The core adaptation algorithm invokes these rules in the order in which they are listed above and exits as soon as one rule *triggers* an adaptation. The idle rule simply garbage-collects unused replicas and enforces a constant refreshment. This avoids that the system gets permanently stuck in some suboptimal configuration. The replication rule applies a pressure on the system such that replications happen until each replica covers at most a network area of radius ρ. This creates an overall number of replicas that depends on ρ and on the diameter of the network, and effectively leads to the division of the network into cells. The migration rule forces replicas to move to locations where the magnitudes of the incoming message flows are in balance. Figure 1 shows how the overall message flow (numbers at the edges) is reduced by migrating towards a dominant flow. This global behavior is not directly coded into the rules. It *emerges* as a number of replicas apply the rules and interact indirectly with one another.

4.2 Service Discovery and Lookup

The instances of the ASG lookup service (LS) [7] are distributed over an ASG network. An ASG network is clustered, and each ordinary node has at least one cluster head in its direct neighborhood. Cluster heads are used to run base services (like the LS) in the ASG. Each LS instance holds location information for all active service replicas. Due to the distributed nature, updates have to be propagated among the LS instances. Since

replica migrations may be executed frequently, and since the transmission bandwidth in the wireless network is the most important resource in the ASG, we refrain from using flooding updates. Instead, we employ a lazy propagation strategy.

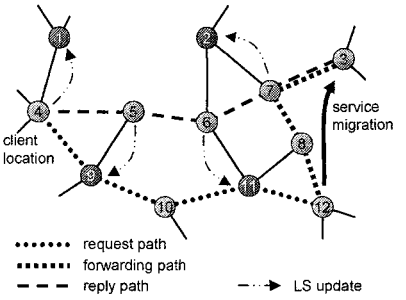


Figure 2. Request-driven update process.

Figure 2 depicts the update process. Only those messages are flooded that announce the creation or the removal of a replica. All other updates (when a replica migrates) are piggybacked by normal service reply messages. Nodes inspect each message that they relay. If a node finds a piggybacked LS update, it informs its cluster head who applies the update to its service table. In this way, updates are propagated *lazily* in those regions that are involved in the sending or the relaying of requests. All other regions remain outdated. However, this does not void the validity of the lookup information due to a second mechanism that cooperates with this lookup process: Every time a replica migrates, it leaves a *forward pointer* at its old location. If a request arrives at this location, it is forwarded, possibly over several former locations of the replica, until it arrives at its current node. The replica sends a reply together with an LS update back to the client. This mechanism ensures that even outdated lookup information is still sufficient to deliver a request correctly. The lazy updating mechanism requires minimal message overhead since updates are only made if needed, and only one dedicated update message is sent to each LS instance along the reply paths.

4.3 Data Consistency

Due to the possible dynamics in an ASG, preserving the consistency in a group of replicas requires an optimistic approach. Replicas may be temporarily separated due to network partitions, or they may be spontaneously created or removed. In order to cope with these circumstances, we have extended the well-known Bayou anti-entropy protocol [13]. The new *Bounded Divergence Group Anti-Entropy Protocol* (BD-GAP) can run reconciliation processes among an arbitrary group of replicas, ensuring eventual consistency. Each replica may autonomously start a reconciliation process and synchronize with the fellow replicas currently known to it. It gets this information from the lookup service. Conflicts due to concurrent initiations of reconciliations are resolved automatically. The protocol chooses the order in which to exchange updates with other

replicas such that the amount of data that has to be exchanged is minimized. Furthermore, it limits the degree to which data stores diverge in such a way that complete state transfers, as they are necessary in Bayou, are completely avoided.

The BD-GAP introduces a stronger coupling than the original anti-entropy protocol by exploiting the nature of the ASG. But still, it allows for enough decoupling to preserve the increased availability introduced by the original protocol. It is adaptive to the degree of dynamics in the system since it gradually reverts to the original pair-wise protocol when the dynamics in the system increases. In situations with low dynamics, it exploits the fact that most replicas are accessible in order to do reconciliations more thoroughly which increases the overall consistency.

4.4 Additional Results

In addition to the results highlighted above, we have also designed and implemented MESHMdl [4], a core middleware on top of which our algorithms and protocols are implemented. MESHMdl is based on mobile agents and the tuple space paradigm to allow for decoupled and asynchronous communication.

In order to show why and in which way the proposed solutions are self-organizing, we created a novel model for *Self-Organizing Software Systems* (SOSS) [8]. The SOSS model can be used to classify existing software systems in order to show whether they are in the class of SOSS or not. Such a model did not exist before which has lead to a growing confusion about the nature of existing systems. We envision that it will serve as an ordering tool beyond the scope of our concrete work, and that the classification of software systems will provide new insights into their general nature.

5 Conclusions and Future Work

In our work, we have made the first step towards a self-organizing infrastructure for ambient services. The three mechanisms that we presented above lay the foundation for this development by providing a self-contained set of functionalities that is required by AmI environments in order to offer facility-specific services to mobile users. We have shown how a self-organized service placement, a self-organized lookup system, and an adequate consistency protocol can be modeled. Furthermore, we have implemented these concepts on top of a set of simple middleware abstractions to proof their validity. Our SOSS model allows us to argue precisely why and how the resulting system is self-organizing.

In the future, more sophisticated systems can be built on these fundamental mechanisms. For example, we have not touched the topics of security and privacy in an ASG environment. These are both essential ingredients to commercially exploitable systems. Moreover, the extension of the ASG with gateways to the Internet and the federation of multiple ASG remain open issues that will be in the focus of our future work.

References

1. G. Cabri, L. Ferrari, L. Leonardi, and F. Zambonelli. The LAICA project: supporting ambient intelligence via agents and ad-hoc middleware. In *Proceedings of the 14th IEEE Inter-*

national Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, pages 39–44, June 2005.
2. K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman. Scenarios for Ambient Intelligence in 2010. Technical Report, The IST Advisory Group (ISTAG), 2001.
3. Michael Hellenschmidt. Distributed Implementation of a Self-Organizing Appliance Middleware. In Gérard Bailly, editor, *Proceedings of sOc-EUSAI 2005 (Smart Objects Conference)*, pages 201–206, 2005.
4. Klaus Herrmann. MESHMdl – A Middleware for Self-Organization in Ad hoc Networks. In *Proceedings of the 1st International Workshop on Mobile Distributed Computing (MDC'03)*, May 2003.
5. Klaus Herrmann. *Self-Organizing Infrastructures for Ambient Services*. PhD thesis, Berlin University of Technology, July 2006. (publication pending).
6. Klaus Herrmann, Kurt Geihs, and Gero Mühl. Ad hoc Service Grid – A Self-Organizing Infrastructure for Mobile Commerce. In *Proceedings of the IFIP TC8 Working Conference on Mobile Information Systems (MOBIS 2004)*. IFIP – International Federation for Information Processing, Springer-Verlag, September 2004.
7. Klaus Herrmann, Gero Mühl, and Michael A. Jaeger. A Self-Organizing Lookup Service for Dynamic Ambient Services. In *25th International Conference on Distributed Computing Systems (ICDCS 2005)*, pages 707–716, Piscataway, NJ, USA, June 2005. IEEE Computer Society Press.
8. Klaus Herrmann, Matthias Werner, and Gero Mühl. A Methodology for Classifying Self-Organizing Software Systems. In *International Conference on Self-Organization and Autonomous Systems in Computing and Communications (SOAS'2006)*, September 2006.
9. Valérie Issarny, Daniele Sacchetti, Ferda Tartanoglu, Françoise Sailhan, Rafik Chibout, Nicole Lévy, and Angel Talamona. Developing Ambient Intelligence Systems: A Solution based on Web Services. *Automated Software Engineering*, 12(1):101–137, 2005.
10. Gregory M. P. O'Hare, Michael J. O'Grady, Rem W. Collier, Stephen Keegan, Donal O'Kane, Richard Tynan, and David Marsh. Ambient Intelligence Through Agile Agents. In Yang Cai, editor, *Ambient Intelligence for Scientific Discovery*, volume 3345 of *Lecture Notes in Computer Science*, pages 286–310. Springer-Verlag, 2004.
11. Marcela Rodríguez, Jesus Favela, Alfredo Preciado, and Aurora Vizcaíno. An Agent Middleware for Supporting Ambient Intelligence for Healthcare. In *Second Workshop on Agents Applied in Health Care (ECAI 2004)*, August 2004.
12. F. Rousseau, J. Oprescu, L.-S. Paun, and A. Duda. Omnisphere: A Personal Communication Environment. In *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS-36 2003)*, 2003.
13. D. B. Terry, M. M. Theimer, Karin Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser. Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System. In *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*, pages 172–182, New York, NY, USA, 1995. ACM Press.
14. M. Vallée, F. Ramparany, and L. Vercouter. A Multi-Agent System for Dynamic Service Composition in Ambient Intelligence Environments. In *Advances in Pervasive Computing, Adjunct Proceedings of the Third International Conference on Pervasive Computing (Pervasive 2005)*, May 2005.

Bereitstellung von Dienstgüte für aggregierte Multimedia-Ströme in lokalen 'Broadcast'-Netzen

Stephan Heckmüller

Dpt. Informatik, Univ. Hamburg, Vogt-Kölln-Str. 30, 22527 Hamburg,
heckmueller@informatik.uni-hamburg.de

Zusammenfassung. Mit der zunehmenden Bedeutung von echtzeitkritischen Audio- und Videoapplikationen wächst die Wichtigkeit von Dienstgütegarantien in Rechnernetzen. Als eine Möglichkeit Dienstgüte zu gewährleisten, bietet sich die Ressourcenreservierung, wobei sich eine statische Reservierung als ineffizient herausgestellt hat. In der vorgestellten Arbeit wurde ein dynamisches Reservierungssystem auf seine Eignung für aggregierte Audio- und Videolasten untersucht. Durch analytische Untersuchungen konnte zunächst gezeigt werden, dass sich mit zunehmender Aggregation die Abschätzung der zu erwartenden Last verbessert. Hierzu wurde der bei der Lastschätzung entstehende Fehler für reale Lasten und verschiedene stochastische Beschreibungsmethoden untersucht. Die Untersuchung des Gesamtsystems zeigte weiterhin, dass sich die Effizienz in fast allen Fällen durch Aggregation erhöhen lässt.

1 Einleitung und Problemstellung

In den vergangenen Jahren ist die Zahl von netzbasierten Multimedia-Anwendungen stetig angewachsen. Ein weiterer Zuwachs für die nächsten Jahre ist zu erwarten. Viele dieser Anwendungen zeichnen sich durch Echtzeitanforderungen aus, womit sich auch die Ansprüche erhöhen, die an ein Kommunikationssystem hinsichtlich der Dienstgüte gestellt werden.
Neben der Dienstgüte in weltweiten Netzen wie dem Internet stellt sich hierbei auch die Frage, wie hinreichende Echtzeitbedingungen in lokalen Netzen gewährleistet werden können. Dies gilt insbesondere dann, wenn diese eine vergleichsweise geringe Bandbreite aufweisen, wie es für die in den letzten Jahren immer mehr zum Einsatz kommenden drahtlosen *WLAN*-Netze der Fall ist. Zur Bereitstellung von Dienstgüte bieten sich hier eine Reihe von Mechanismen an, von denen die Reservierung von Ressourcen in der vorzustellenden Arbeit betrachtet wurde.
Bezüglich der Anwendungsgebiete ist momentan insbesondere eine verstärkte Nutzung von Video- und Sprachanwendungen zu beobachten. Im vorliegenden Beitrag soll die Konzentration daher der Übertragung von Daten, wie sie von solchen Anwendungen induziert werden, gelten.
Weiterhin existieren eine Reihe von Szenarien (Video- bzw. Sprach-Konferenzen,

Streaming-Anwendungen etc.), in denen eine zusammenfassende Reservierung von mehreren Strömen denkbar und sinnvoll ist. Diesen *aggregierten* Strömen soll dabei das Hauptaugenmerk gelten, was auch durch die Tatsache motiviert ist, dass diese im Allgemeinen günstigere Lastcharakteristiken aufweisen als separate Ströme. In Bezug auf den hier betrachteten Reservierungsalgorithmus soll insbesondere untersucht werden, wie sich das Verhalten der Einzelkomponenten und des Gesamtsystems bei wachsendem Aggregationsgrad ändert.
Der Beitrag ist wie folgt strukturiert: Abschnitt 2 gibt zunächst einen Überblick über verwandte Arbeiten. Abschnitt 3 beginnt mit der Beschreibung der verwendeten Schätzfunktionen und bewertet diese dann im Hinblick auf ihre Tauglichkeit für die Schätzung aggregierter Ströme. Weiterhin wird die Schätzung verschiedener stochastischer Prozesse untersucht. In Abschnitt 4 erfolgt hierauf aufbauend die Untersuchung der Ressourcenvergabe unter Nutzung von Schwellwertsystemen. Es wird darüber hinaus eine neue Klasse solcher Systeme vorgestellt, die die (Wahrscheinlichkeits-)Verteilung der Last[1] berücksichtigen.

2 Verwandte Arbeiten

Mit der Veröffentlichung des *IEEE*-Standards *802.11e* zeichnet sich ein verstärktes Interesse an der Bereitstellung von Dienstgüte in lokalen Netzen ab. Hierbei gilt die Konzentration sowohl der Ressourcenreservierung [RPD04,ANT04] als auch der Priorisierung [GDG+03,RNT03] durch unterschiedliche Kanalzugriffsparameter.
Zum Thema Lastschätzung finden sich eine Vielzahl von Vorschlägen, aufgrund welcher Messungen die Lastschätzung vorzunehmen sei. Während in [CF98] die Schätzung ausgehend von der bei jeder Paketankunft aktualisierten Datenrate berechnet wird, wird die Schätzung in vielen Arbeiten aufgrund der beobachteten Warteschlangenlänge unternommen [RPD04,ANT04]. Darüber hinaus kann die Schätzung aber auch direkt ausgehend von der in einem Zeitintervall gemessenen Last vorgenommen werden [WWL05].
Bezüglich der Berechnung der Schätzung aus den gewonnenen Messwerten sind insbesondere die *arithmetische* [Rat91] bzw. die *geometrische Gewichtung* [CF98] zu erwähnen. Darüber hinaus finden noch eine Reihe weiterer Verfahren, wie Kalman-Filter, Anwendung [ASU03]. In der Mehrzahl der mit diesem Thema befassten Arbeiten wird die Schätzung aus der Gesamtmenge der Messdaten gewonnen; demgegenüber wird in [CPZ02] der Ansatz verfolgt, diese nur anhand einer zufällig ausgewählten Teilmenge zu berechnen. (*Random-Sampling*).

3 Lastschätzung und Lastaggregation

Um eine dynamische Ressourcenvergabe vorzunehmen, ist es notwendig, den tatsächlichen Ressourcenbedarf möglichst genau zu prognostizieren. Dazu werden im hier vorgeschlagenen Verfahren zwei Klassen von Schätzfunktionen verwendet [WWL05].

[1] Hier wie im folgenden die während eines Intervalls übergebene Datenmenge.

Arithmetischer Schätzer Die arithmetische Schätzung, zu einem Zeitpunkt t_i, für die gegenwärtig anfallende Last $\hat{\rho}_w(t_i)$ sei definiert wie in Formel (1) angegeben. Die den Schätzwerten zugrundeliegenden Messwerte ρ_i sollen hierbei periodisch zu den Zeitpunkten $t_i := t_0 + \Delta t \cdot i$ zur Verfügung stehen und geben die im Intervall $[t_{i-1}, t_i)$ angefallene Last an. Die Messwerte ρ_i werden im Folgenden als auf das Intervall $[0, 1]$ normiert angenommen.

$$\hat{\rho}_w(t_i) = C_0 \sum_{j=0}^{w-1} \frac{w-j}{w} \rho_{i-j}, \quad C_0 = \frac{2}{w+1}, \quad w \in \mathbb{N} \tag{1}$$

Geometrischer Schätzer Die geometrische Schätzung, zu einem Zeitpunkt t_i, für die gegenwärtig anfallende Last $\hat{\rho}_w(t_i)$ sei definiert wie in Formel (2) angegeben. Die Voraussetzungen bezüglich der Messwerte ρ_i entsprechen den eben getroffenen.

$$\hat{\rho}_\alpha(t_i) = (\alpha \cdot \rho_i) + (1-\alpha) \cdot \hat{\rho}_\alpha(t_{i-1}), \quad \alpha \in [0, 1] \tag{2}$$

Um die Güte der Schätzung in Abhängigkeit von Aggregationsgrad und Lastzusammensetzung untersuchen zu können, gilt es, zunächst eine geeignete Vergleichsmetrik C bereitzustellen. Für eine Folge von Messwerten $\rho = \{\rho_1, ..., \rho_n\}$ und den dazugehörigen Schätzungen $\hat{\rho} = \{\hat{\rho}_1, ..., \hat{\rho}_n\}$ sei diese Metrik definiert wie folgt:

$$C(\hat{\rho}, \rho) = \frac{1}{n} \sum_{i=1}^{n} |\rho_k - \hat{\rho}_k| \tag{3}$$

Die Entwicklung des Schätzfehlers bei der Schätzung von Videolasten[2] für eine Reihe von geometrischen und aritmetischen Schätzern ist in Abbildung 1 dargestellt. Es lässt sich erkennen, dass im Falle des geometrischen Schätzers die Abhängigkeit vom Parameter α mit steigendem Aggregationsgrad n fast vollständig verschwindet. Demgegenüber induzieren für geringe n trägere Schätzer einen kleineren Fehler. Eine qualitativ ähnliche Entwicklung des mittleren Fehlers ist für die betrachteten arithmetischen Schätzer im rechten Teil von Abbildung 1 zu konstatieren. Allerdings zeigt sich hier bereits bei geringen Aggregationsgraden eine wesentlich geringere Sensitivität gegenüber der Wahl des Parameters w. Erst für $n > 10$ sind Vorteile für die trägeren Schätzer auszumachen.

Analoge Untersuchungen wurden ebenfalls für Sprachlasten nach dem Standard *G.711* vollzogen.[3] Hierbei stellte sich wie im Falle der Schätzung von Videolasten eine Verringerung des Schätzfehlers mit steigendem Aggregationsgrad ein. Im Gegensatz zur Schätzung von Videolasten war jedoch eine Erhöhung des Schätzfehlers mit wachsender Einbeziehung der Historie zu verzeichnen. Dies ist auf die *On-Off*-Charakteristik von Sprachlasten zurückzuführen und spricht i.A.

[2] Die hier zugrundeliegende Lasten ergaben sich durch die sukzessive Überlagerung von *MPEG-4*-Traces.

[3] Hierbei fand ein Modell mit negativ exponentiell verteilten Sprach- und Ruhepausen nach [HL86] Verwendung.

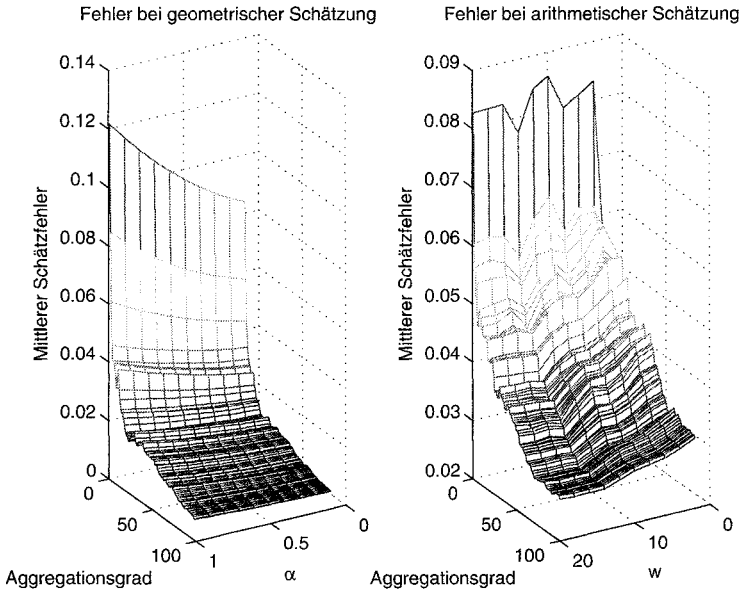


Abb. 1. Schätzfehler in Abhängigkeit vom Aggregationsgrad für Videoübertragungen ($\alpha \in \{0,1; 0,2; ...; 1\}$, $w \in \{1; 2; ...; 20\}$)

für den Einsatz hochempfindlicher Schätzer.
Die so durch die Analyse konkreter Lastmischungen erzielten Ergebnisse lassen sich weiterhin durch die Betrachtung stochastischer Prozesse und der dazugehörigen Schätzung untermauern. Dies soll im folgenden kurz umrissen werden, wobei für eine ausführlichere Darstellung auf [Hec06] verwiesen werden muss:

Unabhängige Verteilungen Ist die Last, die während eines Intervalls induziert wird, unabhängig und identisch mit Dichte $f(x)$ verteilt, so ergibt sich die Dichte der geometrischen Schätzung wie in Formel (4) dargestellt.

$$f_\alpha(x) = f\left(\frac{x}{(1-\alpha)^{N-1}\alpha}\right) * ... * f\left(\frac{x}{(1-\alpha)^0\alpha}\right) \tag{4}$$

Hierauf aufbauend lassen sich die Momente des Schätzfehlers mit Hilfe der Momente der Lastverteilung ausdrücken. Insbesondere ist hervorzuheben, dass die Varianz des Schätzfehlers sowohl bei geometrischer (s. Formel (5)) als auch bei arithmetischer Schätzung (s. Formel (6)) mit der Varianz der Lastverteilung sinkt, was die zuvor erzielten Ergebnisse bestätigt.

$$\sigma_E^\alpha = \left(1 + \frac{\alpha}{2-\alpha}\right)\sigma. \tag{5}$$

$$\sigma_E^w = \left(1 + C_0^2 \cdot \left(\frac{1}{3}w + \frac{1}{2} + \frac{1}{6}w^{-1}\right)\right)\sigma \tag{6}$$

Autoregressive Prozesse Auch für hohe Aggregationsgrade stellt die Unabhängigkeitsannahme eine u.U. unzulässige Abstraktion dar. Um die vorhandene Autokorrelation in die Betrachtung zu integrieren, lässt sich bei gegebener autoregressiver Last eine gleichwertige Beschreibung des Schätzers berechnen, die zeigt, dass mit abnehmender Autokorrelation des Lastprozesses auch die des Schätzprozesses zurückgeht [Hec06].

4 Schwellwertbasierte Ressourcen-Reservierung

Aufbauend auf der im vorhergehenden Abschnitt beschriebenen Lastschätzung wollen wir nunmehr die eigentliche Ressourcen-Reservierung vornehmen. Um Oszillationen und zu häufige Zustandswechsel zu vermeiden, benutzen wir hierbei ein Schwellwertsystem, wie in Def. 1 dargestellt [WWL05].

Definition 1. *Ein* n-Schwellwertsystem *wird als Tupel* $TS(S, \vartheta)$ *definiert mit der Zustandsmenge* $S = \{S_1, S_2, ..., S_n\}$ *und der Übergangsmatrix*

$$\vartheta = \begin{pmatrix} \vartheta_{1,2} & \cdots & \vartheta_{1,n} \\ \vdots & \ddots & \vdots \\ \vartheta_{n,2} & \cdots & \vartheta_{n,n} \end{pmatrix} \in (0,1]^n \times (0,1]^{n-1}$$

mit $\forall k: \quad i < j \rightarrow \vartheta_{k,i} < \vartheta_{k,j} \wedge i < j \rightarrow S_i < S_j$. *Bei gegebenem Zustand* S_i *und der Lastschätzung* $\hat{\rho}$ *ergibt sich der Folgezustand durch*

$$r(S_i, \hat{\rho}) := S_{max_{j=2,...,n}\{\vartheta_{i,j} \leq \hat{\rho}\}}$$

Hierbei stellen die Einträge $\vartheta_{ij}, \quad i < j$ *Aufwärtsschwellwerte dar, alle anderen Einträge repräsentieren Abwärtsschwellwerte.*

Im folgenden soll der in [WWL05] vorgeschlagene Reservierungsalgorithmus auf seine Eignung für aggregierte Lasten untersucht werden. Hierbei finden die *äquidistanten n-Schwellwertsysteme* Anwendung, wie in Def. 2 dargestellt [WHW05].

Definition 2. *Ein* äquidistantes n-Schwellwertsystem $E_n^{b,m}$ *wird definiert als*

$$E_n^{b,m} = TS(S, \vartheta^E)$$

mit der Menge von Zuständen $S_i = 1 - (n-i) \cdot \frac{1-m}{n-1}$ *und der Menge der Transitionen*

$$\vartheta_{i,j}^E = \begin{cases} S_{j-1} - b & , \forall i \geq j \\ S_j - \left(b + \frac{1}{2} \cdot \frac{1-m}{n-1}\right) & , sonst \end{cases}$$

Die Untersuchung erfolgte hierbei für eine Vielzahl von Kombinationen aus Schwellwertsystemen und Schätzern unter Nutzung von Lastmischungen wie sie bereits in Abschnitt 3 verwendet wurden. Exemplarisch sollen die Ergebnisse hier für die sukzessive Überlagerung von Sprachströmen dargestellt werden. Umfangreichere Untersuchungen sind in [Hec06] zu finden.

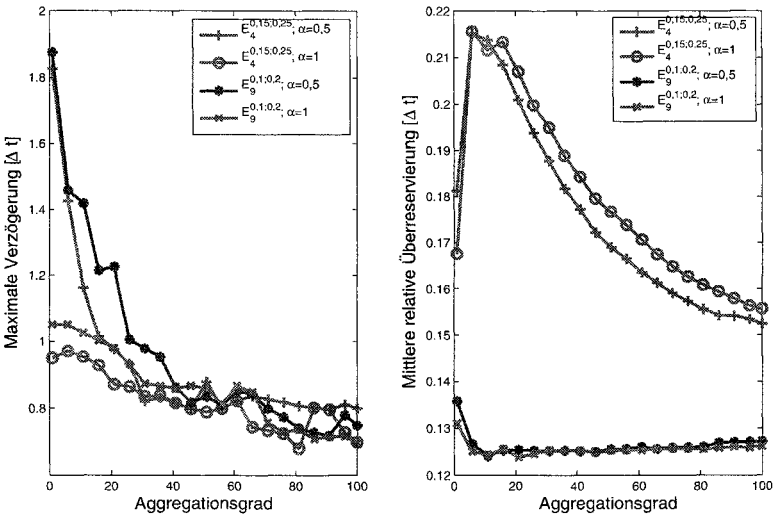


Abb. 2. Verzögerung, Auslastung und Anzahl der Kontrollnachrichten in Abhängigkeit vom Aggregationsgrad für verschiedene Schätzer und Schwellwertsysteme bei Sprachübertragung

Wie in Abbildung 2 ersichtlich wird, nimmt die maximale Verzögerung mit zunehmender Aggregation durchgängig ab, wobei die maximale Verzögerung bei Verwendung des sensitiveren Schätzers ($\alpha = 1$) geringer ist. Die bessere Auslastung der Ressourcen wird weiterhin durchgängig mit dem Schwellwertsystem $E_9^{0,1;0,2}$ erreicht, so dass zusammenfassend dessen Verwendung in Verbindung mit einem hochsensitiven Schätzer angeraten scheint.
Im Gegensatz zu den allgemein verwendbaren $E_n^{b,m}$-Schwellwertsystem beziehen die im folgenden zu untersuchenden Ψ-Schwellwertsysteme (s. Def. 3) die Lastverteilung mit ein. Dies ist für die hier vorgenommene Betrachtung aggregierter Lasten von besonderem Interesse. Da diese sich mit zunehmender Aggregation immer mehr der Normalverteilung annähern, müssen unter dieser Annahme nur Mittelwert und Varianz geschätzt werden, um zu einem parametrisierten Schwellwertsystem zu gelangen.

Definition 3. *Ein* Ψ-verteiltes n-Schwellwertsystem $\Psi_n^{b,m}$ *wird für eine Zufallsvariable mit Verteilungsfunktion* $F_\psi(x)$ *definiert als*

$$\Psi_n^{b,m} = TS(S, \vartheta^\Psi)$$

mit der Menge von Zuständen $S_i = 1 - (n-i) \cdot \frac{1-m}{n-1}$ *und der Menge der Transitionen*

$$\vartheta_{i,j}^\Psi = \begin{cases} S_{j-1} - (1 - F_\psi(S_{j-1})) \cdot b & , \forall i \geq j \\ S_{j-1} - (1 - F_\psi(S_{j-1})) \cdot \left(b - \frac{1}{2} \cdot \frac{1-m}{n-1}\right) & , sonst \end{cases}$$

In Abbildung 3 ist der Vergleich verschiedener Systeme für den Fall einer Videoübertragung dargestellt. Es erfolgte der Vergleich der Ψ-Modelle mit den korrespondierenden E-Modellen für die Aggregationsgrade $n = \{1, 2, ..., 100\}$, wobei die Ergebnisse gemittelt wurden. Insbesondere für die Modelle mit 9 Zuständen lässt sich ein deutlicher Effizienzgewinn feststellen. Demgegenüber ist im Falle der 4-Zustands-Modelle wegen der groben Aufteilung des Zustandsraumes kein Unterschied zwischen beiden Modellklassen zu konstatieren.

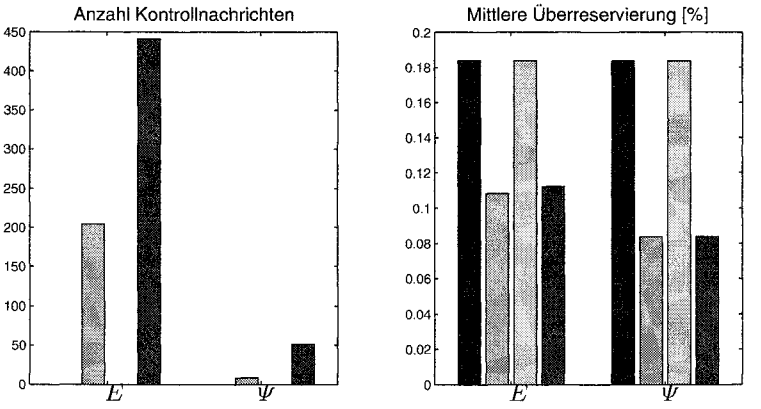


Abb. 3. Vergleich der Kennwerte der verwendeten Ψ- und E-Modelle (E-Modelle von links nach rechts: $E_4^{0,15;0,25}, \alpha = 0,1$; $E_4^{0,15;0,25}, \alpha = 0,3$; $E_9^{0,1;0,2}, \alpha = 0,1$; $E_9^{0,1;0,2}, \alpha = 0,3$, Ψ-Modelle von links nach rechts: $\Psi_4^{0,15;0,25}, \alpha = 0,1$; $\Psi_4^{0,15;0,25}, \alpha = 0,3$; $\Psi_9^{0,1;0,2}, \alpha = 0,1$; $\Psi_9^{0,1;0,2}, \alpha = 0,3$)

5 Resümee und Ausblick

Im vorliegendem Beitrag wurde ein Verfahren zur Bereitstellung von Dienstgüte in lokalen Netzen auf seine Anwendbarkeit für aggregierte Ströme untersucht. Es konnte durch die Untersuchung realer Lasten und stochastischer Prozesse gezeigt werden, dass mit wachsendem Aggregationsgrad die zur dynamischen Ressourcenbereitstellung notwendige Lastschätzung mit zunehmend geringem Fehler möglich ist.

Weiterhin konnte gezeigt werden, dass die Dienstgüte, welche aggregierte Ströme bei Einsatz des untersuchten schwellwertbasierten Verfahrens erfahren, im Vergleich zu nicht-aggregierten Strömen zum Teil deutlich besser ist. Gleichzeitig kann mit Hilfe des verwendeten Reservierungsansatzes ein höherer Gesamtdurchsatz innerhalb des Netzes gewährleistet werden. Darüber hinaus wurde ein Schwellwertsystem vorgeschlagen, mit dem sich – bei bekannter Lastverteilung – die Effizienz weiter erhöhen lässt.

Literaturverzeichnis

[ANT04] Pierre Ansel, Qiang Ni, and Thierry Turletti, An Efficient Scheduling Scheme for IEEE 802.11e, IEEE Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks Workshop (WiOpt), 2004.

[ASU03] T. Anjali, C. Scoglio, and G. Uhl, A new scheme for traffic estimation and resource allocation for bandwidth brokers, Computer Networks **41** (2003), no. 6, 761–777.

[CF98] David D. Clark and Wenjia Fang, Explicit allocation of best-effort packet delivery service, IEEE/ACM Trans. Netw. **6** (1998), no. 4, 362–373.

[CPZ02] Baek-Young Choi, Jaesung Park, and Zhi-Li Zhang, Adaptive random sampling for load change detection, Proceedings of the 2002 ACM SIGMETRICS (New York, USA), ACM Press, 2002, 272–273.

[GDG+03] Priyank Garg, Rushabh Doshi, Russell Greene, Mary Baker, Majid Malek, and Xiaoyan Cheng, Using IEEE 802.11e MAC for QoS over Wireless, The Proceedings of the 22nd IEEE International Performance Computing and Communications Conference (IPCCC 2003), 2003.

[Hec06] Stephan Heckmüller, Bereitstellung von Dienstgüte für aggregierte Multimedia-Ströme in lokalen 'Broadcast'-Netzen, April 2006, Diplomarbeit, Dept. Informatik, Universität Hamburg.

[HL86] H. Heffes and D. M. Lucantoni, A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance, IEEE Journal on Sel. Areas in Comm. **4** (1986), no. 6, 856–868.

[Rat91] Erwin P. Rathgeb, Modeling and Performance Comparison of Policing Mechanisms for ATM Networks., IEEE Journal on Selected Areas in Communications **9** (1991), no. 3, 325–334.

[RNT03] Lamia Romdhani, Qiang Ni, and Thierry Turletti, Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks, IEEE Wireless Communications and Networking Conference (New Orleans), 2003.

[RPD04] Naomi Ramos, Debashis Panigrahi, and Sujit Dey, Dynamic Adaptation Policies to Improve Quality of Service of Multimedia Applications in WLAN Networks, ICST/IEEE Intl. Workshop on Broadband Wireless Multimedia, 2004.

[WHW05] Jürgen Wolf, Stephan Heckmüller, and Bernd Wolfinger, Dynamic Resource Reservation and QoS Management in IEEE 802.11e Networks, Proc. of the Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS) (Philadelphia, USA), vol. 37, July 2005, 149–161.

[WWL05] Bernd Wolfinger, Jürgen Wolf, and Gwendal Le Grand, Improving Node Behaviour in a QoS Control Environment by Means of Load-dependent Resource Redistributions in LANs, International Journal of Communication Systems **18** (2005), no. 4, 373–394.

Ein heuristisches Verfahren zur dienstgütebasierten Optimierung flexibler Geschäftsprozesse

Michael Spahn

Multimedia Communications Lab (KOM), Technische Universität Darmstadt,
Merckstrasse 25, D-64283 Darmstadt

Zusammenfassung. Im Zusammenhang mit der unternehmensübergreifenden Integration externer Partner in Geschäftsprozesse, haben serviceorientierte Architekturen zunehmend an Bedeutung gewonnen. Mit zunehmender Existenz von Servicen gleicher oder ähnlicher Funktionalität steigt die Relevanz nicht-funktionaler Eigenschaften als Selektionskriterium bei der Serviceauswahl. Geschäftsprozesse derart automatisiert aus Servicen zu orchestrieren, dass hierbei definierte Präferenzen und Restriktionen hinsichtlich ihrer nicht-funktionalen Eigenschaften bestmöglich erfüllt werden, führt zu einem Kompositionsproblem, welches sich als NP-schweres Optimierungsproblem formulieren lässt. Gegenstand dieses Beitrages ist die Vorstellung eines heuristischen Verfahrens zur Lösung des Service-Kompositionsproblems, welches mit geringer Laufzeit nahezu optimale Lösungen erzeugt, sowie gut mit der Problemgröße skaliert. Aufgrund der hohen Performanz eignet sich das Verfahren in besonderem Maße für zeitkritische Anwendungen.

1 Einleitung

Services sind abgeschlossene, unabhängige Komponenten, die eine klar definierte Funktionalität anbieten und stellen die elementaren Bausteine einer serviceorientierten Architektur (SoA) dar. Services sind lose gekoppelt und kommunizieren durch den Austausch von Nachrichten. Aufgrund ihrer losen Kopplung können Services zur Laufzeit ersetzt werden und ermöglichen hierdurch ein Höchstmaß an Flexibilität, z.B. im Zusammenhang mit der Integration externer Partner in Geschäftsprozesse.

Neben der Ausführung einer klar definierten Funktionalität, bestehen ebenso Ansprüche in Bezug auf die nicht-funktionalen Eigenschaften, die ein Geschäftsprozess bei seiner Ausführung aufweist. Nicht-funktionale Eigenschaften eines Services oder Geschäftsprozesses werden im Folgenden unter dem Begriff der *Dienstgüte* subsumiert (im Englischen als *Quality of Service* oder *QoS* bezeichnet). Der Begriff der Dienstgüte wird in einem erweiterten Sinne verwendet und umfasst sowohl Parameter der technischen Dienstgüte, wie z.B. Ausführungszeit, Verfügbarkeitswahrscheinlichkeit oder maximaler Durchsatz, als auch nicht-technische Parameter wie z.B. geldliche Kosten eines Service-Aufrufes.

Wird eine Teilaufgabe eines Geschäftsprozesses (Prozessschritt) durch einen Service realisiert, so lässt sich dieser Prozessschritt prinzipiell durch alle funktional identischen Services ausführen. Existieren für jeden Prozessschritt mehrere, funktional identische Services, so ergibt sich im Hinblick auf die Ausführung eines Geschäftsprozesses eine hohe Anzahl verschiedener Zuordnungsmöglichkeiten von Servicen zu

Prozessschritten (*Service-Kompositionen*), die in einer funktional identischen Ausführung des Geschäftsprozesses resultieren. Die Dienstgüte eines Geschäftsprozesses variiert jedoch mit der konkreten Service-Komposition, die zu seiner Ausführung verwendet wird. In einem derartigen Szenario ist es erstrebenswert, zum Ausführungszeitpunkt eines Geschäftsprozesses automatisiert immer genau jene Service-Komposition zu wählen, welche definierte Ansprüche an die Dienstgüte bestmöglich erfüllt. Ansprüche können in Form von Präferenzen (z.B. Minimierung der Kosten und der Fehlerrate) und Restriktionen (z.B. Ausführungszeit darf 30 Sekunden nicht überschreiten) formuliert werden. Das Problem Services anhand ihrer Dienstgüte derart für eine Service-Komposition auszuwählen, dass alle bzgl. eines Geschäftsprozesses formulierten Restriktionen eingehalten und zugleich alle formulierten Präferenzen bestmöglich erfüllt werden, wird im Folgenden als *Service-Kompositionsproblem* bezeichnet.

Das Service-Kompositionsproblem lässt sich als NP-schweres Optimierungsproblem formulieren, dessen exakte Lösung aufgrund der Komplexität sehr zeitintensiv ist. Um die Ermittlung der optimalen Service-Komposition zur Laufzeit sinnvoll realisieren zu können, ist es erforderlich die benötigte Rechenzeit auf ein praktikables Maß zu reduzieren. Zum Zwecke der zeiteffizienten Lösung des Service-Kompositionsproblems, wurde ein heuristisches Lösungsverfahren entwickelt, welches anstelle der optimalen Lösung lediglich eine sehr gute zulässige Lösung bestimmt. Dem Verlust an Genauigkeit steht jedoch ein erheblicher Laufzeitvorteil im Vergleich zu exakten Lösungsmethoden gegenüber. Die präsentierte Arbeit kann als konsequente Fortsetzung der, mit der Entwicklung der dienstgüteunterstützenden Web-Service-Architektur *WSQoSX (Web Services Quality of Service Architectural Extension)* [1, 2] begonnen Arbeiten gesehen werden.

Der Rest dieses Beitrages gliedert sich wie folgt: In Abschnitt 2 erfolgt die Formalisierung des Service-Kompositionsproblems durch Formulierung eines Optimierungsmodells. Ein heuristisches Verfahren zur Lösung des entwickelten Optimierungsmodells wird in Abschnitt 3 vorgestellt. Ergebnisse der Evaluierung dieses Verfahrens werden in Abschnitt 4 präsentiert. In Abschnitt 5 werden verwandte Arbeiten behandelt, bevor in Abschnitt 6 ein abschließendes Fazit gezogen wird.

2 Modell des Service-Kompositionsproblems

In diesem Abschnitt wird eine Formalisierung des Service-Kompositionsproblems vorgenommen. Die Modellierung erfolgt in Form eines gemischt-ganzzahligen linearen Optimierungsproblems.

2.1 Service-Komposition

Service-Komposition bezeichnet den Vorgang der Auswahl und Verbindung von Servicen zur Vorbereitung der Ausführung eines Geschäftsprozesses. Das Ergebnis der Service-Komposition ist ein *Ausführungsplan* des Geschäftsprozesses. Der Ausführungsplan legt fest, welcher Prozessschritt durch welchen Service realisiert wird. Im Folgenden wird die Betrachtung auf rein sequentielle Geschäftsprozesse beschränkt.

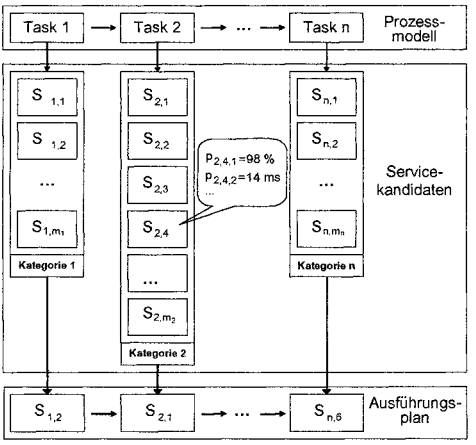


Abb. 1. Service-Komposition

Abbildung 1 zeigt ein aus *n Tasks* bestehendes sequentielles Prozessmodell. Jeder Task spezifiziert hierbei abstrakt die in einem Prozessschritt benötigte Funktionalität (z.B. Bonitätsprüfung). In einem *sequentiellen Prozess* wird Task i ($i=1,...,n$) genau dann vor Task i' ($i'=1,...,n$) ausgeführt, wenn $i<i'$. Die Menge der m_i verschiedenen Services ($S_{i,j}$), welche die in Task i benötigte Funktionalität bereitstellen, wird *Kategorie i* genannt. Für jeden Service in jeder Kategorie wird eine binäre Variable $x_{i,j}$ eingeführt, wobei $x_{i,j}=1$ bedeutet, dass in Kategorie i der Service j zur Ausführung im Ausführungsplan ausgewählt ist. Um sicherzustellen, dass nur je ein Service pro Task ausgewählt wird, soll gelten:

$$\sum_{j=1}^{m_i} x_{i,j} = 1 \quad \forall i = 1,\ldots,n. \tag{1}$$

2.2 Dienstgütemodell

Die Dienstgüte von Servicen lässt sich durch eine beliebige Anzahl (k) einzelner Parameter beschreiben. Um die Dienstgüte des Prozesses aus den Dienstgüteparametern der zur Ausführung ausgewählten Services berechnen zu können, wird eine *Aggregationsfunktion* für jeden Dienstgüteparameter benötigt. $p_{i,j,k}$ bezeichnet den Wert des Dienstgüteparameters mit Index k des Services j in Kategorie i. Die Aggregation der Dienstgüteparameter der Services zu einem Gesamtdienstgüteparameter des Prozesses ist abhängig vom Typ des jeweiligen Dienstgüteparameters. Im verwendeten Optimierungsmodell werden drei Dienstgüteparameter berücksichtigt: Additive Parameter (p^+), multiplikative Parameter (p^*) und Minimaloperator-Parameter (p^{min}). Beispielhaft zur Verwendung sei angeführt, dass sich die Gesamtausführungszeit eines Prozesses additiv durch Summation der Einzelausführungszeiten der ausgeführten Services ergibt, wohingegen sich die Verfügbarkeitswahrscheinlichkeiten multiplikativ zur Gesamtverfügbarkeitswahrscheinlichkeit aggregieren und sich der mögliche Gesamtdurchsatz durch Anwendung des Minimaloperators auf die Menge der

Durchsätze der ausgeführten Services ergibt. Die Aggregationsfunktionen der drei berücksichtigten Parametertypen können Tabelle 1 entnommen werden.

Tabelle 1. Aggregationsfunktionen der Parametertypen

	Parameter	Gesamtparameterwert
Additive Parameter	$p_{i,j}^{+}$	$x^{+} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} p_{i,j}^{+} x_{i,j}$
Multiplikative Parameter	$p_{i,j}^{\bullet}$	$x^{\bullet} = \prod_{i=1}^{n} \sum_{j=1}^{m_i} p_{i,j}^{\bullet} x_{i,j} \approx 1 - \sum_{i=1}^{n} \left(1 - \sum_{j=1}^{m_i} p_{i,j}^{\bullet} x_{i,j} \right)$
Minimaloperator-Parameter	$p_{i,j}^{\min}$	$x^{\min} = \underset{i=1}{\overset{n}{Min}} \left(\sum_{j=1}^{m_i} p_{i,j}^{\min} x_{i,j} \right)$

2.3 Restriktionen

Gesamtdienstgüteparameter können durch *Restriktionen* beschränkt werden, wie z.B. durch die Bedingung, dass die Gesamtausführungszeit des Prozesses 30 Sekunden nicht überschreiten darf. Ein Ausführungsplan zu einem Prozess ist genau dann *gültig*, wenn er alle definierten Restriktionen erfüllt. Restriktionen können durch die Beschränkung der Gesamtdienstgüteparameter in Form von Nebenbedingungen im Optimierungsmodell ausgedrückt werden.

2.4 Optimierungskriterien

Präferenzen bezüglich der vorzunehmenden Optimierung des Prozesses werden durch eine Gewichtung der Gesamtdienstgüteparameter formuliert. Ein Gewicht w definiert hierbei, welcher Zusatznutzen durch die Verbesserung eines Gesamtdienstgüteparameters um eine Einheit erzeugt wird. Die Zielfunktion $F(\vec{x})$ bewertet die Dienstgüte des Prozesses in Form des erzeugten Gesamtnutzens als gewichtete Summe der Gesamtdienstgüteparameter. Werden die Gesamtdienstgüteparameter entsprechend ihres Typs gruppiert, so lässt sich die Zielfunktion wie folgt formulieren:

$$F(\vec{x}) = \sum_{l=1}^{k^{+}} w_l^{+} x_l^{+} + \sum_{l=1}^{k^{\bullet}} w_l^{\bullet} x_l^{\bullet} + \sum_{l=1}^{k^{\min}} w_l^{\min} x_l^{\min} \qquad (2)$$

Die Dienstgüte eines Prozesses wird umso besser bewertet, je höher der gestiftete Gesamtnutzen ist. Dadurch entspricht die Optimierung der Dienstgüte einer Maximierung des Zielfunktionswertes. Die *optimale Lösung* ist jene gültige Lösung mit dem maximalen Zielfunktionswert.

3 Heuristik zur Lösung des Service-Kompositionsproblems

Zum Zwecke der zeiteffizienten Approximation der optimalen Lösung wurde die Heuristik H1_RELAX_IP entwickelt. Sie basiert auf mehreren Schritten. In einem ersten Schritt erfolgt eine Relaxation des Optimierungsproblems durch Fallenlassen der Ganzzahligkeitsbedingung der vormals binären Lösungsvariablen $x_{i,j}$, sodass diese jede reele Zahl zwischen 0 and 1 annehmen können. In einem zweiten Schritt erfolgt die exakte Lösung des relaxierten Optimierungsproblems mittels eines Standardverfahrens (z.B. Simplex-Algorithmus), was in zeiteffizienter Weise möglich ist. In einem dritten Schritt wird ein Backtracking-Algorithmus verwendet um eine zulässige Lösung des ursprünglichen, unrelaxierten Optimierungsproblems zu bestimmen.

Die Lösung des relaxierten Optimierungsproblems liefert Anhaltspunkte dafür, welche Services potentiell Teil des optimalen Ausführungsplans sein könnten. Sind beispielsweise die Lösungsvariablen $x_{i,g}$=0,25 und $x_{i,l}$=0,75, so kann abgeschätzt werden, dass in Kategorie i der Service l mit einer höheren Wahrscheinlichkeit Teil des optimalen Ausführungsplans ist, als Service g. Entsprechend dieser Abschätzung werden die Services j in ihrer jeweiligen Kategorie i absteigend nach dem Wert ihrer Lösungsvariablen $x_{i,j}$ sortiert (Abbildung 2). Sind in einer Kategorie mehrere Services mit $x_{i,j}$=0 vorhanden, so werden diese anhand ihres potentiellen Beitrags zur Zielfunktion sortiert. Details hierzu können [4] entnommen werden.

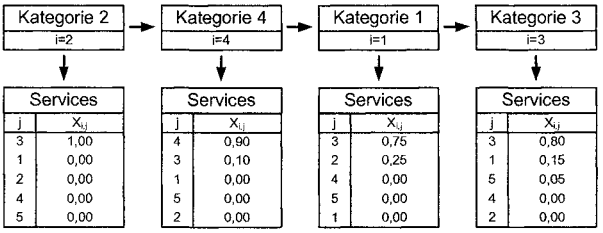


Abb. 2. Sortierung der Services und Kategorien

Zusätzlich zur Sortierung der Services innerhalb einer Kategorie, erfolgt eine aufsteigende Sortierung der Kategorien nach der Anzahl enthaltener Services mit $x_{i,j}>0$. Der Backtracking-Algorithmus zur Bestimmung der zulässigen Lösung beginnt mit der Kategorie, welche die geringste Anzahl von Servicen mit $x_{i,j}>0$ besitzt. Je geringer die Auswahl an Servicen mit $x_{i,j}>0$ in einer Kategorie ist, desto höher die Wahrscheinlichkeit, mit einer Auswahl gemäß der Größe von $x_{i,j}$ eine richtige Entscheidung zu treffen. In Abbildung 2 würde der Backtracking-Algorithmus beispielsweise die Entscheidung für einen Service aus Kategorie 2 vor der Entscheidung in Kategorie 3 treffen, da Kategorie 2 weniger Wahlmöglichkeiten mit $x_{i,j}>0$ anbietet und diese darüber hinaus eine höhere (geschätzte) Wahrscheinlichkeit besitzen Teil der optimalen Lösung zu sein. Werden Entscheidungen zur Platzierung von Services im Ausführungsplan in dieser Reihenfolge getroffen, so werden potentiell falsche Entscheidungen zum Schluss getroffen. Dies erhöht die Performanz des Backtracking-Algorithmus aufgrund der Tatsache, dass die Revision einer falschen Entscheidung umso teurer ist, je früher diese getroffen wurde.

```
i=1;
Exec_Plan={0, 0, ..., 0};
end=false;
while (not end) {
  repeat {
    if (Exec_Plan[i]<mi) Exec_Plan[i]++;
  } until (Exec_Plan is valid or Exec_Plan[i]=mi);
  if (Exec_Plan is invalid) {
    Exec_Plan[i]=0;
    if (i>1) i--; else end=true;
  } else
    if (i<n) i++; else end=true;
}
```

Abb. 3. Pseudocode des Backtracking-Algorithmus

Abbildung 3 skizziert den verwendeten Backtracking-Algorithmus in Form von Pseudocode. Zu Beginn wird der Ausführungsplan geleert und der erste Service aus der ersten Kategorie (gemäß der Sortierreihenfolge) an der entsprechenden Position im Ausführungsplan gesetzt. Im Beispiel aus Abbildung 2 ist dies der Service mit Index $j=3$ in Kategorie 2. Wird hierdurch keine Restriktion verletzt, wird der erste Service der nächsten Kategorie gesetzt. Im Falle der Verletzung einer Restriktion wird der aktuelle Service solange durch den nächsten Service der aktuellen Kategorie ersetzt, bis keine Restriktion mehr verletzt wird. Ist dies nicht möglich, wird die aktuelle Position des Ausführungsplans geleert und die Ersetzung in der vormals behandelten Kategorie fortgesetzt. Sobald keine Restriktionen mehr verletzt werden, wird wieder zur nächsten Kategorie fortgeschritten. Konnten alle Positionen des Ausführungsplans erfolgreich besetzt werden, so ist eine zulässige Lösung für das unrelaxierte Optimierungsproblem gefunden. Durch die Berücksichtigung der Lösung des relaxierten Optimierungsproblems in Form der Vorsortierung der Kategorien und Services, entstehen hierbei sehr schnell nahezu optimale Lösungen.

4 Evaluierung

Zum Zwecke der Evaluierung wurden mittels eines eigens implementierten Datengenerators Service-Kompositionsprobleme (Testfälle) erzeugt. Diese wurden einerseits exakt durch das Verfahren Brach&Bound mittels des Solvers lp_solve (http://lpsolve.sourceforge.net/5.1/) gelöst und andererseits approximativ durch Anwendung der entwickelten Heuristik. Hierbei wurde ein Vergleich der benötigten Berechnungsdauern durchgeführt, sowie die Genauigkeit der Approximation der optimalen Lösung (Lösungsgüte) durch die Heuristik erfasst. Um den Einfluss verschiedener Parameter auf die Performanz und Lösungsgüte der Heuristik aufzuzeigen, wurden Mengen von Testfällen erzeugt, in denen jeweils ein Parameter gezielt variiert wurde. Es wurde der Einfluss folgender Parameter analysiert: i.) Prozesslänge (Anzahl der Tasks des Prozesses), ii.) Kandidatenanzahl (Anzahl der Services in einer Kategorie) und iii.) Restriktionsstärke (Stärke der Einschränkung des zulässigen Lösungsraums durch definierte Nebenbedingungen). Da die Resultate aufgrund ihres Umfanges an dieser Stelle nicht umfassend vorgestellt werden können, wird im Folgenden lediglich der Einfluss der Prozesslänge auf H1_RELAX_IP verkürzt und exemplarisch dargestellt. Eine umfassendere Darstellung der Resultate kann [3] und [4] entnommen werden.

4.1 Einfluss der Prozesslänge auf H1_RELAX_IP

Um den Einfluss der Prozesslänge zu analysieren wurde diese in sieben Schritten von drei auf 21 Tasks erhöht und jeweils eine Testmenge mit 35 Testfällen erstellt. Die Kandidatenanzahl betrug hierbei konstant 40 Services pro Kategorie, wobei jeder Service durch vier Dienstgüteparameter beschrieben wurde und jeder der vier Gesamtdienstgüteparameter mittels einer Restriktion beschränkt wurde. Tabelle 2 können die Mittelwerte der durchgeführten Messungen entnommen werden. Mit zunehmender Prozesslänge weist die Heuristik gegenüber dem Solver einen stetig steigenden und erheblichen Vorteil in Bezug auf die Berechnungsdauer auf. H1_RELAX_IP skaliert gut mit steigender Prozesslänge und weist eine Lösungsgüte auf hohem Niveau aus, die sich nur marginal mit steigender Prozesslänge verschlechtert. So erreicht H1_RELAX_IP beispielsweise im Falle einer Prozesslänge von 21 Tasks 98,83% des Zielfunktionswertes der optimalen Lösung, benötigt zur Berechnung jedoch nur 0,19% der Berechnungsdauer der exakten Lösung durch den Solver.

Tabelle 2. Ergebnisse der Variation der Prozesslänge

Tasks	Solver Avg: t_s [ms]	H1_RELAX_IP Avg: t_h [ms]	Avg: $\frac{t_h}{t_s}$	Avg: $\frac{F(\vec{x}_h)}{F(\vec{x}_s)}$
3	4,2864	5,2734	136,08%	99,96%
6	16,3458	8,4682	69,17%	99,89%
9	194,9850	13,9468	22,63%	99,72%
12	1.062,7156	20,3617	8,86%	99,44%
15	5.976,9378	26,2847	3,43%	99,38%
18	60.772,1917	35,5667	0,92%	99,23%
21	264.177,5668	43,3683	0,19%	98,83%

5 Verwandte Arbeiten

Die Problematik, eine dienstgüteoptimierte Service-Komposition zeiteffizient ermitteln zu können, wird auch in anderen Arbeiten adressiert. Zeng et al. stellen in [5] die Middleware-Plattform *„AgFlow"* vor, die eine dienstgütebasierte Service-Komposition durch die exakte Lösung eines ganzzahligen linearen Optimierungsproblems (GLOP) realisiert. Yu und Lin befinden dieses Verfahren jedoch in [6] als zu zeitintensiv für zeitkritische Szenarien und stellen die *„QoS-capable Web Service Architecture" (QCWS)* vor, in der sie ein vereinfachtes Modell sowohl in Form eines multidimensionalen Multiplechoice-Knapsack-Problems (MMKP) als auch in Form eines mehrfach restringierten Optimalen-Wege-Problems (MCOP) mittels heuristischer Ansätze lösen. Canfora et al. verfolgen in [7] einen Ansatz auf Basis genetischer Algorithmen. Aufgrund der Kürze des Beitrags sei an dieser Stelle für eine umfassendere Darstellung verwandter Arbeiten auf [3] und [4] verwiesen.

6 Fazit und Ausblick

In diesem Beitrag wurde ein heuristisches Verfahren zur Lösung des dienstgütebasierten Service-Kompositionsproblems vorgestellt. Das heuristische Verfahren verwendet einen Backtracking-Algorithmus auf Basis der Lösung eines relaxierten gemischt-ganzzahligen linearen Optimierungsproblems zur Erzeugung eines optimierten

Ausführungsplans für Prozesse, die aus Servicen orchestriert werden. Durch eine Evaluation des Verfahrens konnte gezeigt werden, dass es mit einer sehr geringen Berechnungsdauer nahezu optimale Lösungen erzeugen kann. Das Verfahren benötigt eine wesentlich geringere Berechnungsdauer als herkömmliche Verfahren auf Basis der exakten Lösung (gemischt-)ganzzahliger linearer Optimierungsprobleme und skaliert gut mit zunehmender Problemgröße, was eine Verwendung des Verfahrens in zeitkritischen Szenarien erlaubt.

Als Ausblick sei an dieser Stelle auf das Potential der Erweiterung des vorgestellten Verfahrens zum Zwecke der Realisierung effizienter Replanning-Strategien verwiesen [8]. Mittels Replanning erfolgt eine Adaption des Ausführungsplans derart, dass auch bei einem vom Plan abweichenden Verhalten der Services die Optimalität des Ausführungsplans stets gewährleistet bleibt.

7 Anmerkungen

Dieser Beitrag basiert auf der Diplomarbeit des Verfassers, die am Fachgebiet Multimedia Kommunikation (KOM) der Technischen Universität Darmstadt im Rahmen des E-Finance Lab e.V. (Cluster 2) angefertigt wurde. Teile der Arbeit wurden bereits unter [3] und [4] veröffentlicht. Für ihre Unterstützung möchte sich der Verfasser an dieser Stelle bei Dipl.-Wirtsch.-Inform. R. Berbner, Dr.-Ing. O. Heckmann und Prof. Dr.-Ing. R. Steinmetz bedanken.

8 Literatur

1. Berbner R., Heckmann O. und Steinmetz R.: „An Architecture for a QoS driven composition of Web Service based Workflows", Networking and Electronic Commerce Research Conference (NAEC 2005), Gardasee, Italien, 2005.
2. Spahn M., Berbner R., Heckmann O., et al.: „WSQoSX - Eine dienstgütebasierte Web-Service-Architektur", Technical Report KOM 2004-07 (V2), Technische Universität Darmstadt, 2005.
3. Berbner R., Spahn M., Repp N., et al.: „Heuristics for QoS-aware Web Service Composition", IEEE 4th International Conference on Web Services (ICWS 2006), Chicago, USA, 2006.
4. Spahn M., Berbner R., Heckmann O., et al.: „Ein heuristisches Optimierungsverfahren zur dienstgütebasierten Komposition von Web-Service-Workflows", Technical Report KOM 02-2006, Technische Universität Darmstadt, 2006.
5. Zeng L., Benatallah B., Ngu A., et al.: „QoS-aware Middleware for Web Service composition", IEEE Transactions on Software Engineering, 30 (2004) 5, S. 311-328.
6. Yu T. und Lin K.-J.: „A Broker-Based Framework for QoS-Aware Web Service Composition", International Conference on e-Technology, e-Commerce, and e-Services (EEE 2005), Hong Kong, China, 2005.
7. Canfora G., Penta M. D., Esposito R., et al.: „An approach for QoS-aware service composition based on genetic algorithms", Genetic and Evolutionary Computation Conference (GECCO 2005), Washington DC, USA, 2005.
8. Berbner R., Spahn M., Repp N., et al.: „QoS-aware Replanning of Web Service Workflows", angenommen zur IEEE International Conference on Digital Ecosystems and Technologies (DEST 2007), Cairns, Australien, 2007.

Index der Autoren

Abeck, Sebastian, 101
Abendroth, Dirk, 251
Angrishi, Kishore, 189

Basin, David, 225
Baumgart, Ingmar, 139
Bessler, Sandford, 77
Biermann, Jürgen, 101
Binzenhöfer, Andreas, 15
Braun, Torsten, 27
Buschmann, Carsten, 151

Dreibholz, Thomas, 39

Emig, Christian, 101

Fischer, Stefan, 151

Gabner, Rene, 77
Gaitzsch, Martin, 267
Gamer, Thomas, 275
Götze, Joachim, 89
Gross, James, 291
Gross, Julia, 77
Guenkova-Luy, Teodora, 63

Haenni, Rolf, 213
Hauck, Franz J., 63
Hauser, Ralf, 225
Heckmüller, Stephan, 307
Heissenbüttel, Marc, 259
Hellbrück, Horst, 151
Henjes, Robert, 113
Herrmann, Klaus, 299
Hillenbrand, Markus, 89
Himmler, Valentin, 113
Hof, Hans-Joachim, 139

Jonczy, Jacek, 213

Kaiser, Bruno, 225
Kangasharju, Jussi, 51
Killat, Ulrich, 189
Kohlas, Reto, 213
Kraft, Daniel, 163

Lahde, Sven, 283
Langer, Kim, 101
Lehrieder, Frank, 177
Lombriser, Clemens, 127

Mäder, Andreas, 201
Menth, Michael, 113, 177
Milbrandt, Jens, 177
Mühlhäuser, Max, 51
Müller, Paul, 89

Oppliger, Rolf, 225

Rathgeb, Erwin P., 39
Ries, Sebastian, 51
Rodenhaeuser, Aldo, 225
Roggen, Daniel, 127

Schäfer, Günter, 3, 163
Scheidegger, Matthias, 27
Schlosser, Daniel, 113
Schmidt, Holger, 63
Schnabel, Holger, 15
Schöller, Marcus, 237
Spahn, Michael, 315
Staehle, Dirk, 201
Stäger, Mathias, 127
Strufe, Thorsten, 3

Tröster, Gerhard, 127

Völker, Lars, 237

Werner, Christian, 151
Wildhagen, Jens, 3

Zeiss, Joachim, 77
Zhang, Ge, 89
Zhang, Shu, 189
Zitterbart, Martina, 139

Printing: Mercedes-Druck, Berlin
Binding: Stein + Lehmann, Berlin