Network Architectures
And Services
NET 2009-10-1

# IITM & SN SS09

**Proceedings of the Seminars**
**Innovative Internet Technologies and Mobile Communications (IITM)**
**Sensor nodes - Operation, Network and Application (SN)**
Summer Semester 2009

Munich, Germany, 16.04.2009 - 27.07.2009


**Editors**

Georg Carle, Corinna Schmitt, Uwe Baumgarten

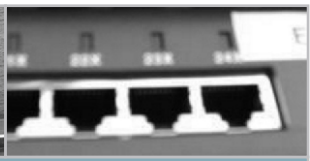
**Organisation**

Chair for Network Architectures and Services
Chair for Operating Systems and System Architectures
Department of Computer Science, Technische Universität München


Technische Universität München







Network Architectures
and Services
NET 2009-10-1

# IITM & SN SS09

## Proceedings zu den Seminaren
## Innovative Internettechnologien und Mobilkommunikation (IITM)
## Sensorknoten – Betrieb, Netze und Anwendungen (SN)

München, 16.04.2009-27.07.2009

Editoren: Georg Carle, Corinna Schmitt, Uwe Baumgarten

Organisiert durch den Lehrstuhl Netzarchitekturen und Netzdienste (I8)
und den Lehrstuhl für Betriebssysteme und Systemarchitektur (I13),
Fakultät für Informatik, Technische Universität München

Technische Universität München

IITM & SN SS09
Seminar: Innovative Internettechnologien und Mobilkommunikation
Seminar: Sensorknoten – Betrieb, Netze und Anwendungen

Editors:

Georg Carle
Lehrstuhl Netzarchitekturen und Netzdienste (I8)
Technische Universität München
D-85748 Garching b. München, Germany
E-mail: carle@net.in.tum.de
Internet: http://www.net.in.tum.de/~carle/

Corinna Schmitt
Lehrstuhl Netzarchitekturen und Netzdienste (I8)
Technische Universität München
D-85748 Garching b. München, Germany
E-mail: schmitt@net.in.tum.de
Internet: http://www.net.in.tum.de/~schmitt/

Uwe Baumgarten
Lehrstuhl Betriebssysteme und Systemarchitektur (I13)
Technische Universität München
D-85748 Garching b. München, Germany
E-mail: baumgaru@in.tum.de
Internet: http://www13.in.tum.de/

Cataloging-in-Publication Data

IITM & SN SS09
Proceedings zu den Seminaren “Innovative Internettechnologien und Mobilkommunikation” und „Sensorknoten – Betrieb, Netze und Anwendungen“
München, Germany, 16.04.2009 – 27.07.2009
Georg Carle, Corinna Schmitt, Uwe Baumgarten
ISBN: 3-937201-07-6

ISSN: 1868-2634 (print)
ISSN: 1868-2642 (electronic)
Lehrstuhl Netzarchitekturen und Netzdienste (I8) NET 2009-10-1
Series Editor: Georg Carle, Technische Universität München, Germany

# Vorwort

Wir präsentieren Ihnen hiermit die Proceedings zu den Seminaren “Innovative Internettechnologien und Mobilkommunikation” (IITM) und „Sensorknoten – Betrieb, Netze und Anwendungen“ (SN), die im Sommersemester 2009 an der Fakultät Informatik der Technischen Universität München stattfanden.
Im Seminar IITM wurden Beiträge zu unterschiedlichen Fragestellungen aus den Gebieten Internettechnologien und Mobilkommunikation vorgestellt. Die folgenden Themenbereiche wurden abgedeckt:

- SNMP - Simple Network Management Protocol
- Peer-to-Peer TV
- Wie verändert man das Internet?
- Patente - Einführung, wirtschaftliche Bedeutung von Patenten auf Kommunikationsprotokolle
- Webservices for embedded systems
- Delay Tolerant Networks
- Identity Management
- Spiele in P2P Systemen
- Tor und Angriffe gegen Tor
- Anwendung herkömmlicher Peer-to-Peer Konzepte für verteilte Datenbanken
- Evolution der Kernnetze im Mobilfunk
- IT-Sicherheit - Faktor Mensch und psychologische Aspekte (interdisziplinäres Thema)

Im Seminar SN wurden Vorträge zu verschiedenen Themen im Forschungsbereich Sensorknoten vorgestellt. Die folgenden Themenbereiche wurden abgedeckt:

- Überblick und Vergleich von Sensorknotentechnologien
- TinyOS: Ein Betriebssystem für Sensorknoten
- Einführung in das Structural Health Monitoring

Wir hoffen, dass Sie den Beiträgen dieser Seminare wertvolle Anregungen entnehmen können. Falls Sie weiteres Interesse an unseren Arbeiten habe, so finden Sie weitere Informationen auf unserer Homepages http://www.net.in.tum.de und http://www13.in.tum.de.

München, Oktober 2009

Georg Carle

Corinna Schmitt

Uwe Baumgarten

# Preface

We are very pleased to present you the interesting program of our main seminars on “Innovative Internet Technologies and Mobil Communication” (IITM) and “Sensor nodes – Operating, Network and Application” (SN) which took place in the summer semester 2009.

In the seminar IITM we deal with issues of Internet technologies and mobile communication. The seminar language was German, and the majority of the seminar papers are also in German. The following topics are covered by this seminar:

- SNMP – Simple Network Management Protocol
- Peer-to-Peer TV
- How can we change the Internet?
- Patents – Introduction, economic importance of patents for communication protocols
- Webservices for embedded systems
- Delay Tolerant Networks
- Identity Management
- Games in Peer-to-Peer systems
- Tor and attacks against Tor
- The Application of Common Peer-to-Peer Concepts in the Field of Distributed Databases
- Evolution of core networks in mobile communication
- IT-Security – Human impact and psychological factors (interdisciplinary topic)

In the seminar SN talks to different topics in current research tasks in the field of sensor nodes were presented. The seminar language was German, and also the seminar papers. The following topics are covered by this seminar:

- Overview and comparison of sensor technologies
- TinyOS: An operating system for sensor nodes
- Introduction to Structural Health Monitoring

We hope that you appreciate the contributions of these seminars. If you are interested in further information about our work, please visit our homepages http://www.net.in.tum.de and http://www13.in.tum.de.

Munich, October 2009

# Seminarveranstalter

### Lehrstuhlinhaber

Georg Carle, Uwe Baumgarten, Technische Universität München, Germany

### Seminarleitung

Corinna Schmitt, Technische Universität München, Germany

# Betreuer

Bernhard Amann, *Technische Universität München, DFG Emmy Noether Research Group Member*
Tobias Bandh, *Technische Universität München, Wiss. Mitarbeiter I8*
Ali Fessi, *Technische Universität München, Wiss. Mitarbeiter I8*
Marc Fouquet, *Technische Universität München, Wiss. Mitarbeiter I8*
Nils Kammenhuber, *Technische Universität München, Wiss. Mitarbeiter I8*
Holger Kinkelin, *Technische Universität München, Wiss. Mitarbeiter I8*
Andreas Müller, *Technische Universität München, Wiss. Mitarbeiter I8*
Heiko Niedermayer, *Technische Universität München, Wiss. Mitarbeiter I8*
Marc-Oliver Pahl, *Technische Universität München, Wiss. Mitarbeiter I8*
Corinna Schmitt, *Technische Universität München, Wiss. Mitarbeiterin I8*

# Kontakt:

{carle,schmitt,bandh,braun,elser,fessi,fouquet,kinkelin,mueller,muenz,heiko,pahl}@net.in.tum.de
baumgaru@in.tum.de

# Seminarhomepage

http://www.net.in.tum.de/de/lehre/ss09/seminare/

# Inhaltsverzeichnis

## Seminar Innovative Internet-Technologien und Mobilkommunikation

### Session 1: Peer-to-Peer



### Session 2: Sicherheit und Patente



### Session 3: Netzwerkarchitekturen und Protokolle

## Seminar Sensorknoten – Betrieb, Netze und Anwendungen

# Peer-to-Peer TV

Matthias Hellerer
Betreuer: Ali Fessi
Seminar Innovative Internettechnologien und Mobilkommunikation SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: hellerer@cs.tum.edu

***Kurzfassung*— Mit der zunehmenden Verbreitung von Breitbandinternetanschlüssen in privaten Haushalten steigt auch die Nachfrage nach Multimediainhalten im Internet. Dies bedeutet für die Anbieter solcher Inhalte eine starke Zunahme des Netzwerkverkehrs und verursacht hohe Kosten. Gerade bei der Verbreitung grosser Datenmengen an viele Endnutzer haben sich Peer-to-Peer Netzwerke seit Jahren bewährt. Hier soll nun ein Überblick darüber gegeben werden, wie sie auch im Bereich des Media-Streaming eingesetzt werden können.**

***Schlüsselworte*— Peer-to-peer, P2P, TV, Video, Media, Streaming**

## I. Einleitung

Seit nun mehr als 50 Jahren ist das Fernsehen das bedeutendste Massenmedium [1]. Während das Internet zunehmend an Bedeutung gewinnt werden auch hier verschiedene Multimediaangebote, insbesondere Videos, verstärkt nachgefragt. Sowohl in Form von Einzelvideos als auch als Fernsehsender über die Internetverbindung. Ermöglicht wird dieser Trend vorallem durch die zunehmende Verbreitung von breitbandigen Internetanschlüssen im privaten Bereich. Die meisten Fernsehesender bieten heute ihr Programm zumindest teilweise auch im Internet an und Videoplattformen wie zum Beispiel `youtube` - die weltweit meist frequentierte - liefern schon heute mehr als ein hundert millionen Videos am Tag aus [2]. Zu dem möchten auch immer mehr Anbieter von Kabelfernsehen, Telefonanschlüssen und Internetverbindungen ihren Markt ausbauen, in dem sie verschiedenen Dienstleistungen zu bündeln versuchen. Beliebt ist hier vorallem die Kombination aus Internet, Telefon und Fernsehen. Aus Kostengründen wird es auch hier angestrebt, all diese Dienste über eine Verbindung zu liefern [3].

Im de facto Aufbau des Internets, mit Einzelverbindungen für jeden Videoabruf, werden hierfür gigantische Bandbreiten benötigt. Bei einer durchschnittlichen Videodateigrösse von 35MB kann das Transfervolumen von `youtube` auf etwa 40GB/s geschätzt werden, wobei diese Zahl linear mit der Anzahl der angefragten Videos steigt. Es werden daher in Zukunft neue Technologien benötigt um die Belastung der Server zu reduzieren. Es liegt nahe hierfür *Peer-to-Peer* Netzwerke einzusetzen, da sie bei grossen Datenmengen deutlich besser skalieren und sie sich schon seit langem bei der Verteilung von grosser Datenmengen an viele Nutzer bewährt haben.

In diesem Artikel soll nun zunächst geklärt werden, warum sich die für diesen Zweck ursprünglich vorgesehene Technologie des *IP-Multicast* nicht durchsetzen konnte, dann wird darauf eingegangen, wie *Peer-to-Peer*-Netzwerke für *Media-Streaming* eingesetzt werden können. Abschliessend werden zwei verfügbare Programme vorgestellt und offene technische Fragen angesprochen.

## II. Peer-to-Peer im Vergleich zu IP-Multicast

Für die Verteilung von grossen immer gleichen Daten an viele Endnutzer, in erster Linie für *Media-Streaming* für Radio und Fernsehen, ist eigentlich schon seit über 20 Jahren das *IP-Multicast* nach `RFC1112`, als bisher einzige, grössere Erweiterung des *IP*-Standards, vorgesehn [4]. Dabei wird ein Daten-*Stream* vom Sender nur einmal versendet und auf dem Weg durch das Netzwerk von jedem *Router* je nach Bedarf an mehrere weitere *Router* oder Endnutzer gesendet. Die Verteilung an mehrere Endnutzer erfolgt also durch das Netzwerk selbst. Zu diesem Zweck ermittelt der Nutzer zunächst mittels des *Internet Group Management Protocols* einen nahe gelegenen Router, der auf dem Weg zur Quelle liegt, und teilt diesem den Wunsch mit einer *IP-Multicast Group* beizutreten zu wollen. Dieser versucht dann auf die gleiche Weise selbst die entsprechenden Daten zu erhalten. Dies setzt sich fort, bis die Quelle selbst oder ein *Router*, der über die entsprechenden Daten verfügt, gefunden wurde.

Dennoch konnte sich diese Technik bis heute nicht durchsetzen. Dies kann vor allem darauf zurück geführt werden, dass *IP-Multicast* von den Internetanbietern und damit auch von deren Netzwerken nicht unterstützt wird. Die Gründe hierfür sind zahlreich, am entscheidensten ist aber wohl der folgende: *IP-Multicast* verlangt von jedem *Router* eine Gruppenverwaltung und verletzt damit das Netzwerkarchitekturprinzip des *Stateless Router*, in dem einem *Router* einzig die Aufgabe zukommt jedes Paket, unabhängig von allen anderen zu verteilen und so transparent ausgetauscht oder neu konfiguriert werden kann. Damit in Zusammenhang steht, dass es nicht möglich ist etablierte Systeme zur Garantierung von Sicherheit und Zuverlässigkeit auf dieses Konzept zu übertragen und eigens hierfür spezialisierte Systeme sind nicht verfügbar. Zu erwähnen sind hier in erster Linie *Denial-of-Service*-Attacken, die durch *IP-Multicast* besonders begünstigt werden. Insgesamt steigt durch diese Problematik vor allem der Administrationsaufwand und damit auch die Kosten für die

Netzwerkbetrieber erheblich, weshalb die meisten Netzwerke, sowohl von *Internet Backend Providern*, Internetanbietern und lokale Netzwerke, *IP-Multicast* nicht unterstützen [3], [5].

Im Gegensatz dazu sind *Peer-to-peer*-Netzwerke deutlich einfacher zu installieren und sie sind nicht auf eine Unterstützung von Seiten den Netzwerkes angewiesen, sondern basieren auf *Client*-seitigen Programmen, die Einzelverbindungen zu mehreren anderen, gleichartigen Programmen unterhalten und so ein *Overlay*-Netzwerk aufspannen in dem Jeder mit Jedem Daten austauschen kann. Zudem skalieren sie besser, insbesondere, auch im Bezug auf die Anzahl der Medienanbieter nicht nur Mediennutzer [6] und sind zudem variabler wodurch sie besser für Nutzergruppen mit hoher Dynamik geeignet sind [5].

## III. Besonderheiten von Media-Streaming

Die wichtigste Unterscheidung im Bereich des Videostreaming unterteilt die Systeme in zwei Kategorien:

- *Video-on-Demand*, bei dem Videos individuell ausgewählt und zu jeder Zeit angesehen werden können,
- und *Real-Time Broadcasting*, bei dem der selbe Videostream gleichzeitig von mehreren Leuten angesehen wird, ähnlich dem konventionellem Fernsehen.

Auch wenn beide Arten von den meisten derzeit verfügbaren Programmen gleichzeitig unterstützt werden ist der technologische Unterschied nicht zu vernachlässigen.

### A. Besonderheiten des Video-on-Demand

*Video-on-Demand* ähnelt aus technische Sicht den verbreiteten *Peer-to-Peer*-Netzwerken wie beispielsweise `Bittorrent` recht stark. Dennoch sind diese ohne Veränderungen nicht für den Einsatz im Bereich des *Media-Streaming* geeignet, da sie stark auf die Verbreitung kompletter Dateien optimiert sind. Um ein angefordetes Video sofort ansehen zu können, ohne es zunächst komplett herunter laden zu müssen, muss mit dem *Download* der Datei am Anfang angefangen werden. Dies mag selbstverständlich erscheinen, jedoch verfolgen Programme wie `Bittorrent` eine andere Strategie. Sie versuchen, zur Steigerung der Netzwerkeffizenz zunächst die am wenigsten verfügbaren Teile zu erhalten [7], [8]. Die nötigen Änderungen an den etablierten Protokollen sind aber insgesamt eher gering, weshalb auch Programme existieren, die *Video-on-Demand* innerhalb von, ansonsten unveränderten, `Bittorrent`-Netzwerken realisieren.

### B. Besonderheiten des Real-Time-Broadcasting

*Real-Time-broadcasting* unterscheidet sich deutlich von diesem System. Es muss nicht mehr eine Datei, die von den *Peers* in der Regel lange Zeit vorgehalten wird, verteilt werden, sondern lediglich die Videodaten für einen fast instantanen Zeitpunkt. Daraus folgt, dass die Daten bereits kurz nach dem *Download* ihre Gültigkeit verlieren und gelöscht werden können. Das Netzwerk muss somit fortlaufend mit neuen Daten versorgt werden, die innerhalb möglichst kurzer Zeit auf alle *Peers* verteilt werden sollten. Als Datenquellen bieten sich verschiedene Systeme an. So währe es zum Beispiel theoretisch möglich, dass *Peers*, die über eine Möglichkeit zum Rundfunkempfang verfügen, die Datenquellen darstellen. Da dies jedoch zu zahlreichen weiteren Problemen führt, insbesondere die Sicherstellung der Datenintegrität, wird üblicherweise der Ansatz verfolgt, dass besonders priviligierte *Peers* oder *Server* die Netzwerke mit den Daten-*streams* "füttern" und das Netzwerk lediglich zur Weiterverbreitung dieser Daten dient [9].

Dafür hält jeder Teilnehmer einen gewissen Zeitabschnitt vor. Dieser wird zunächst von mehreren anderen *Peers* heruntergeladen und kann dann angesehen und weiterverteilt werden. Je grösser dieser Abschnitt gewählt wird desto effizienter kann er verteilt werden [10]. Jedoch kann jeder *Peer* eine Dateneinheit immer erst dann weiterverteilen, wenn er diese Daten zunächst selbst erhalten hat. Die benötigte Zeit von der Einspeisung bis zum Empfänger ist somit durch die Netzwerktopologie, beziehungsweise die Anzahl von *Hops* auf diesem Weg, und die Länge des vorgehaltenen Zeitabschnitts beschränkt [11]–[13].

Da diese Netzwerke somit schon zentralisiert strukturiert sind und die Verbindungslänge zwischen *Peer* und *Server* von entscheidender Bedeutung ist, werden die *Overlay*-Netzwerke zentralisiert aufgebaut, das heisst nahezu die gesamte Verwaltung läuft über einen oder mehrere zentrale *Server*.

### C. Unterstützung durch Server

Sowohl *Video-on-Demand* als auch *Real-Time-Broadcasting* leiden unter dem Problem, dass die meisten Privathaushalte über *ADSL*-Zugänge (*Asynchronous Digital Subscriber Line*) angeschlossen sind. Das heisst, *Upload* und *Download* verfügen über unterschiedliche Bandbreite. Da der *Upload* in aller Regel deutlich kleiner dimensioniert ist aber jeder Nutzer gerne einen möglichst grossen Teil der ihm zur Verfügung stehenden *Download*-Bandbreite für eine bessere Bildqualität nutzen möchte ist der Einsatz zusätzlicher Server, die versuchen die hieraus resultierende, ungleiche Bilanz zu nivelieren, für die bestmögliche Qualität des Dienstes unumgänglich, da alle Daten, die ein Teilnehmer empfangen möchte, zunächst von einem Anderen verschickt werden müssen [14]. Vorallem aber werden die Server benötigt, um es zu ermöglichen, dass auch wenig nachgefragte Inhalte schnell verteilen zu können, da diese in einem reinen *Peer-to-Peer*-Netzwerk nur auf wenigen oder sogar gar keinen *Peers* vorhanden wären und die Servicequalität beträchtlich unter dieser Einschränkung leiden würde. Im Bereich des Fernsehens werden beispielsweise die beliebtesten $10\%$ der Sender von $80\%$ der Nutzer angesehn [1]. Es gibt also in der Regel wenige sehr stark nachgefragte Inhalte und viele sehr wenig nachgefragte. Vorallem diese sehr stark nachgefragten Inhalten sollen mittels *Peer-to-Peer Media Streaming* effizienter verteilt werden. Da die wenig nachgefragten Inhalte jedoch naturgemäss wenig Netzwerkverkehr verursachen, stellt eine Verteilung über Server im kommerziellen Bereich in der Regel ein kleineres Problem dar [10].

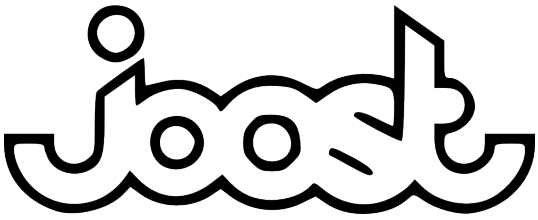
### D. Multiple Description Coding

Desweiteren variiert die den Nutzern zur Verfügung stehende Bandbreite mit unter sehr stark. Um dennoch jedem Nutzer eine für ihn best mögliche Bildqualität liefern zu können wird *Multiple Description Coding* eingesetzt. Bei dieser Technik wird das Video in mehrere Ebenen eingeteilt. Jede Ebene verfügt über eine relativ geringe Datenrate, kann einzeln dekodiert werden und führt zu einem Video mit relativ geringer Qualität. Durch die Hinzunahme weiterer Ebenen und deren Verschmelzung zu einem Video kann die Videoqualität dann schrittweise verbessert werden. Neben einer Anpassung an die individuellen Bandbreitenunterschiede ist das Netzwerk so auch flexibler und kann zum Beispiel den Nutzern schon sehr schnell nach dem Beitreten in ein Netzwerk zumindest ein Bild, wenn auch mit vergleichsweise schlechter Qualität liefern. Dies ist vorallem interesant, da die meisten Nutzer gerne ”zappen” also nur wenige Sekunden in ein Programm rein sehen. So liegt der Median der Betrachtungszeit eines Senders bei gerade ein mal $8s$ [1]. Desweiteren ist es so auch möglich flexibel auf Bandbreitenschwankungen zu reagieren und Systeme auf zu bauen, die Nutzer, die das Netzwerk durch starken *Upload* verbessern, mit höhere Bildqualität zu belohnen. [10], [15]–[17]

## IV. Kurzvorstellung vorhandener Systeme

Hier sollen nun sehr kurz je eines der populärsten Programme für *Video-on-Demand* und *Real-Time-broadcasting* vorgestellt werden. Neben diesen existiert noch eine Vielzahl weiterer Programme, jedoch können diese derzeit eingeteilt werden in freie Programme, die in erster Linie dem akademischen Bereich entspringen und über eine stark eingeschränkte Funktionalität verfügen, und kommerzielle Programme, die zwar meist kostenlos sind oder zumindest der grösste Teil ihrer Funktionen kostenlos ist, über deren Funktionsweise jedoch wenig gesagt werden, da der Quellcode nicht einsichtig ist. Die hier vorgestellten Programme sind beide kommerziell und wurden nach einigen Tests aufgrund ihrer subjektiv besten Qualität für diese Einführung ausgewählt, da in erster Linie die Möglichkeiten der *Peer-to-Peer Media Streaming* Technologie aufgezeigt werden sollen.

### A. Joost



Nach dem Verkauf von `Skype` an `eBay` haben sich die ehemaligen Entwickler auf ein neues Projekt konzentriert. `Joost` bietet seit 2007 ein *Peer-to-Peer* basiertes *Video-on-Demand*-System an, das es ermöglicht auf `www.Joost.com` [18] mehr als 8.000 Videoclips und ganze Kinofilme kostenfrei anzusehen. Finanziert wird dies in erster Linie mit dem Erlös aus dem Verkauf von `Skype`. Das Einstellen eigener Videos ist zum jetzigen Zeitpunkt nicht ohne weiteres möglich. Alle Videos müssen derzeit noch von den Entwicklern zunächst rechtlich und qualitativ bewertet werden und dann von diesen in das System integriert werden [19].

Zunächst als eigenes Programm konzipiert, baut es heute auf dem `Adobe FlashPlayer` auf und ist vollständing in den *Browser* integriert [20], so das es genutzt werden kann, ohne dass das Installieren zusätzlicher Software nötig ist. Im Zuge dieser Umstellung sind derzeit keine *Real-Time-Broadcasts* verfügbar [21]–[23].

### B. Zattoo

`Zattoo` bietet auf `www.Zattoo.com` [24] zum einen eine eigenes Programm, zum anderen eine Webapplikation im Betabetrieb mit stark eingeschränkter Funktionalität an. Im Angebot der schweizer Firma finden sich reguläre, in Deutschland frei empfangbare Fernsehsender. Das verfügbare Programm ist, aus rechtlichen Gründen, vom Aufenthaltsort des Benutzters abhängig. So sind in Deutschland derzeit 115 Sender empfangbar, darunter vorallem öffentlich rechtliche. Desweiteren werden Programme für die Länder Schweiz, Grossbritanien, Spanien und Dänemark angeboten. Das Programm ist wie auch *Joost* properitär, jedoch ist die Benutzung, bis auf die hoch qualitative Version bestimmter Sender, kostenlos. Finanziert wird das Projekt zum einen durch ein kostenpflichtiges Abonement zum Empfang qualitativ höhere Versionen bestimmter Sender, zum anderen durch die Überblendung der Werbepausen mit eigener Werbung.

Der zeitliche Versatz zwischen dem *Peer-to-Peer*-Video von Zattoo im Vergleich zu einem konventionellem Satelitenfernsehen belief sich in eigenen Tests auf ca $4 - 5$s.

## V. Offene Fragen

Trotz des viel versprechenden Ansatzes, bleiben noch einige Fragen ungeklärt.

### A. Wenig optimiertes Routing

Trotz zunehmender Verbreitung von *Peer-to-peer*-Netzwerken, werden die Möglichkeiten bisher noch nicht voll ausgeschöpft. Reale Lokalität wird so gut wie nicht genutzt und das *Routing* der meisten Internetanbiert ist klar auf das Abrufen von Internetseiten in grossen Rechenzentren optimiert. Abhilfe könnte hier die *Overlay-Traffic-Optimization* bringen. Dabei wird versucht die *Routing*-Pfade für *Peer-to-peer*-Netzwerke möglichst kurz zu halten und nach Möglichkeit im selben Netzwerk. Dies ist besonders für die Anbieter wichtig um auch die vorallem mit *Peer-to-Peer*-Verkehr verbundenen hohen Kosten für

Netzwerkverkehr in fremde Netze zu veringern. Die kürzeren *Routing*-Pfade wiederum veringern die Latenzzeiten und erhöhen die Bandbreite des *Peer-to-Peer*-Netzwerkes [25].

### B. Nutzerdynamik

Ein anderes Problem stellt *Churn* dar. Das heisst, dass die meisten Zuschauer eines Programms dieses nur wenige Sekunden betrachten, bevor sie wieder weiterschalten [1]. Vorallem zu Beginn von Sendungen wird so ein beträchtlicher Teil der Zuschauer "ausgewechselt". Für das zu Grunde liegende Netzwerksystem stellt dies ein grosses Problem dar, da es zum einen möglichst schnell einen neuen *Stream* darstellen soll und zum anderen kaum Zeit hat sich zu optimieren [12], [21]–[23].

### C. Rechtliche Lage

Ein weiteres Problem stellt die rechtlich unsichere Lage dar. Während es bei *Video-on-Demand* im Allgemeinen zwei Kategorien gibt - freie Inhalte die auch frei heruntergeladen werden können und kommerzielle Inhalte, die in der Regel gegen Bezahlung heruntergeladen werden können - konzentriert sich *Real-Time-broadcasting* auf Inhalte des konventionellen Fernsehens. In wie weit diese Art der Weiterverwertung mit geltendem Recht vereinbar ist, ist weitgehend ungeklärt. So verfügte Zattoo beispielsweise bis 1. April 2009 über die Rechte zur Nutzung des Programms der deutschen öffentlich-rechtlichen Sender. Seit Ablauf dieses Vertrages wird die Verwendung durch die Sender zwar weiter geduldet, im Mai 2009 jedoch, wurde die Austrahlung von Inhalten der Verleiher Warner Bros und Universal, durch diese, per einstweiliger Verfügung untersagt [26]. Weiter verkompliziert wird die Lage durch die Erhebung von GEZ-Gebühren auf Internetanschlüsse und Eingriffe in den Programmablauf, wie die geänderte Werbung bei Zattoo.

### D. Absicherung

*Peer-to-Peer Media Streaming* Netzwerke sehen sich vielseitigen Angriffen ausgesetzt. Die meisten entsprechen den bekannten Angriffen auf Einzelrechner oder allgemeine *Peer-to-peer*-Netzwerke, jedoch ist *Real-Time-Broadcasting* besonders anfällig für *Pollution-Attacks*, dabei gibt ein Angreifer vor regulärer Teilnehmer des Netzwerkes zu sein, verschickt jedoch fehlerhafte oder komplett zufällige Daten. Ohne ausreichende Absicherung kann dies dazu führen, dass diese Daten von den Empfängern wiederum weiter gegeben werden und auch diese Empfänger dann die eingeschleusten Daten weiter geben, bis somit ein grosser Teil des Netzwerkes mit korrumpierten Daten beliefert wird, wodurch der eigentliche Netzwerkverkehr stark in Mitleidenschaft gezogen wird. Dies kann soweit führen, dass das gesamte Netzwerk unbenutzbar wird.

*Hash Verification*, wie sie aus den meisten *Peer-to-peer*-Netzwerken für Datenaustaush bekannst ist, kann hier nicht angewendet werden, da in der Regel selbst der Sender bis kurz vor der Übertragung die zu sendenden Daten nicht kennt. Andere Abwehrmethoden wären beispielsweise *Blacklisting* oder *Traffic Encryption*, jedoch bieten auch sie keinen sicheren Schutz oder wirken sich anderweitig nachteilig aus. Die derzeit einzige, sichere Technik stellt das *Chunk Singning* dar. Dabei wird jeder einzelne Zeitabschnitt vom Sender mittels eines *Public-Key*-Verfahrens signiert und von den *Peers* erst nach einer Verifikation weitergereicht. Jedoch steigt durch die mitzuführenden Signaturen, die die Dateigrösse an, zu dem verursacht diese Technik sowohl beim Sender als auch beim Empfänger sehr viel CPU-Belastung und da die Verifikation vor dem Weitereichen abgeschlossen sein muss erhöhen sich auch die Latenzzeiten entsprechend [27].

## VI. Zusammenfassung und Ausblick

Der Anteil bandbreitenintensiver Multimediainhalte im Internet wird auch in Zukunft stark zunehmen. Um dies zu ermöglichen müssen andere Wege als die bisher vebreitete *Client-Server*-Technologie beschritten werden. *Peer-to-Peer Media Streaming* stellt hier, meiner Meinung nach, das zukunftsträchtigste, bekannte System dar. Auch wenn noch viel Forschung zur Optimierung benötigt wird und viele rechtliche Fragen noch geklärt werden müssen, zeigen die vorhandenen Programme, trotz ihrer noch eingeschränkten Möglichkeiten, das grosse Potential dieser Technologie.

Darüber hinaus hat diese Technlogie das Potential dazu auch weitergehende Veränderungen zu bewirken. So wäre es beispielsweise mit einem entsprechenden *Peer-to-Peer*-Netzwerk denkbar, dass Privatpersonen ohne grossen finanziellen Aufwand - da keine teueren Sendeanlagen benötigt werden - in die Lage versetzt werden einen eigenen Fernsehsender aufzubauen. Eine Entwicklung analog zu der von *Webradios*, die aufgrund der geringeren Bandbreitenanforderung schon heute verbreitet sind, wäre so möglich [3].

Auf jeden Fall kann davon ausgegangen werden, dass *Peer-to-Peer Media Streaming* schon bald eine bedeutende Rolle bei der Verteilung von Multimediainhanlten im Internet zu kommen wird.

## LITERATUR


[1] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, "Watching television over an IP network," in *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2008, pp. 71–84.

[2] (2009, May) YouTube Fact Sheet. YouTube limited. [Online]. Available: http://www.youtube.com/t/fact_sheet

[3] S. Tewari and S. Menon, "Peer-to-peer streaming: Lowering barrier to entry in IPTV," in *Proc. 2nd International Symposium on Advanced Networks and Telecommunication Systems ANTS '08*, Dec. 15–17, 2008, pp. 1–3.

[4] S. Deering, *RFC1112 - Host extensions for IP multicasting*, http://www.faqs.org/rfcs/rfc1112.html, Network Working Group Std., August 1989.

[5] Y. hua Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1456–1471, Oct. 2002.

[6] A. Ganjam and H. Zhang, "Internet Multicast Video Delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 159–170, Jan. 2005.

[7] A. P. C. da Silva, E. Leonardi, M. Mellia, and M. Meo, "A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems," in *Proc. Eighth International Conference on Peer-to-Peer Computing P2P '08*, Sep. 8–11, 2008, pp. 279–288.

[8] N. Parvez, C. Williamson, A. Mahanti, and N. Carlsson, "Analysis of bittorrent-like protocols for on-demand stored media streaming," in *SIGMETRICS '08: Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 2008, pp. 301–312.

[9] H. Yin, C. Lin, F. Qiu, X. Liu, and D. Wu, "TrustStream: a novel secure and scalable media streaming architecture," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2005, pp. 295–298.

[10] R. Rejaie, "Anyone can broadcast video over the internet," *Commun. ACM*, vol. 49, no. 11, pp. 55–57, 2006.

[11] Y. Liu, "On the minimum delay peer-to-peer video streaming: how realtime can it be?" in *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*. New York, NY, USA: ACM, 2007, pp. 127–136.

[12] F. Covino and M. Mecella, "Design and evaluation of a system for mesh-based P2P live video streaming," in *MoMM '08: Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*. New York, NY, USA: ACM, 2008, pp. 287–290.

[13] D. Lou, Y. Mao, and T. H. Yeap, "The production of peer-to-peer video-streaming networks," in *P2P-TV '07: Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*. New York, NY, USA: ACM, 2007, pp. 346–351.

[14] V. Janardhan and H. Schulzrinne, "Peer assisted VoD for set-top box based IP network," in *P2P-TV '07: Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*. New York, NY, USA: ACM, 2007, pp. 335–339.

[15] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang, "Using layered video to provide incentives in P2P live streaming," in *P2P-TV '07: Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*. New York, NY, USA: ACM, 2007, pp. 311–316.

[16] J. A. Pouwelse, J. R. Taal, R. L. Lagendijk, D. H. J. Epema, and H. J. Sips, "Real-time video delivery using peer-to-peer bartering networks and multiple description coding," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, Oct. 10–13, 2004, pp. 4599–4605.

[17] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–93, Sep. 2001.

[18] (2009, Jun.) Joost. Joost N.V. [Online]. Available: http://www.joost.com/

[19] J. De Boever, "Value Networks of P2P TV: An Analysis of Actors and Their Roles," in *Proc. Third International Conference on Internet and Web Applications and Services ICIW '08*, Jun. 8–13, 2008, pp. 686–695.

[20] (2008, Oct.) Joost launches web-based video service with community features. Joost N.V. [Online]. Available: http://press.joost.com/2008/10/joost_launches_webbased_video.html

[21] M. Alhaisoni, A. Liotta, and M. Ghanbari, "An assessment of Self-managed P2P Streaming," in *2009 Fifth International Conference on Autonomic and Autonomous Systems*, Oct. 2009, pp. 34–39.

[22] M. Alhaisoni and A. Liotta, "Characterization of signaling and traffic in Joost," *Peer-to-Peer Networking and Applications*, vol. 2, pp. 75–83, Apr. 2008.

[23] D. Ciullo, M. Mellia, M. Meo, and E. Leonardi, "Understanding P2P-TV Systems Through Real Measurements," in *Proc. IEEE Global Telecommunications Conference IEEE GLOBECOM 2008*, Nov. 2008, pp. 1–6.

[24] (2009, Jun.) Zattoo - watch online TV. Zattoo Inc. [Online]. Available: http://www.zattoo.com/

[25] W. Kellerer, "Architectural Considerations for the Management of the Future Internet: Management of Overlay Traffic and Virtualized Networks," in *Dagstuhl Seminar 09052*. Landsberger Strasse 312, 80687 Munich, Germany: DOCOMO Communications Laboratories Europe GmbH Ubiquitous Networking Research Group, Jan. 2009.

[26] D. Bouhs, "Bei 'Psycho' bleibt der Bildschirm schwarz," *Frankfurter Rundschau*, May 2009. [Online]. Available: http://www.fr-online.de/in_und_ausland/kultur_und_medien/medien/?em_cnt=1762269&

[27] P. Dhungel, X. Hei, K. W. Ross, and N. Saxena, "The pollution attack in P2P live video streaming: measurement results and defenses," in *P2P-TV '07: Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*. New York, NY, USA: ACM, 2007, pp. 323–328.

# Spiele in Peer-to-Peer Systemen

Michael Schmitt
Betreuer: Bernhard Amann
Seminar Innovative Internet-Technologien und Mobilkommunikation SS 2009
Lehrstuhl Netzarchitekturen und Netzdienste
Institut für Informatik
Technische Universität München
schmitmi@in.tum.de

## Kurzfassung
Massively Multiplayer Online Games (MMOG's) nehmen unter den Computerspielen eine immer wichtigere Rolle ein. Auf Anbieterseite lassen sich durch MMOG's und deren Erfolg erhebliche Gewinne erzielen. Mit dem Erfolg und der daraus folgenden Anzahl der Spieler, der Größe und Komplexität eines Spiels steigt aber gleichzeitig der Kommunikationsaufwand und dadurch wiederum die benötigte Bandbreite. Auch die Hardware auf Serverseite muss dementsprechend großzügig ausgelegt sein, um die jeweilige Spieleranzahl verkraften zu können. Klassische Systeme stoßen deshalb immer wieder an ihre Grenzen[7], weshalb Peer-to-Peer Systeme (P2P) immer wieder als Problemlösung in Betracht gezogen werden. In diesem Paper sollen deswegen sowohl Chancen und Möglichkeiten, als auch Probleme von Peer-to-Peer Systemen im Zusammenhang mit Spielen anhand von Beispielen diskutiert werden.

## Schlüsselworte
Peer-to-Peer, Spiele, Massively Multiplayer Online Games, SimMUD, FreeMMG, pSense

## 1. Einleitung
Durch das enorme Interesse an MMOGs und der zunehmenden Komplexität der Spiele stoßen die Server der Anbieter immer wieder an ihre Grenzen. Auch die Belastung des Internets durch den produzierten Traffic stellte ein Problem sowohl für Anbieter, als auch Kunden dar. Aus diesem Grund rücken P2P-Systeme immer wieder in den Fokus der Forschung. Durch P2P erhofft man sich auf der einen Seite Kosten für Hardware zu sparen, da die Kunden/Clients die Berechnung und Simulation der Spielwelt übernehmen sollen. Auf der anderen Seite möchte man den Traffic einsparen, indem Nachrichten möglichst direkt an die Clients verschickt werden, die sie auch benötigen. Im folgenden Abschnitt wird die Problemstellung MMOG und die Verwaltung einer Spielwelt vorgestellt. In Abschnitt 3 werden die bisherigen Lösungen etwas genauer betrachtet und deren Vor-, sowie Nachteile herausgearbeitet. Abschnitt 4 beschäftigt sich mit der allgemeinen Lösungsstrategie die mit P2P verfolgt wird, während in Abschnitt 5 drei Lösungen im Detail vorgestellt werden. In Abschnitt 6 werden diese Lösungen noch einmal zusammenfassend bewertet. Abschnitt 7 zeigt die Nutzung von P2P in heutigen Spielen, während Abschnitt 8 eine kurze Zusammenfassung und einen kleinen Ausblick in die Zukunft gibt.

## 2. Problemstellung Verwaltung von MMOG
In den letzten Jahren spielen MMOGs und vor allem Massively Multiplayer Online Role-Play Games eine immer wichtigere Rolle unter den Computerspielen. Nicht nur, dass sich immer mehr Spieler für dieses Genre interessieren, auch auf Betreiber-Seite steigt das Interesse. So lassen sich erhebliche Gewinne durch diese Art von Spielen erwirtschaften, da Kunden meist nicht nur für das Spiel selbst bezahlen, sondern Abonnements für Spielzeit abschließen müssen oder in „Cash-Shops" reales Geld gegen virtuelle Spielgegenstände eintauschen können. Aktuelle Beispiele für erfolgreiche MMOGs sind „World of Warcraft" („WORLD OF WARCRAFT zählt jetzt mehr als 11,5 Millionen Abonnenten weltweit"[5]), EVE-Online („EVE-Online: Mehr als 300 000 aktive User"[6]) oder „Herr der Ringe Online" sowie „Age of Conan". Aus technischer Sicht entstehen aber viele Probleme, welche bei „normalen" Multiplayerspielen nicht so stark ins Gewicht fallen.

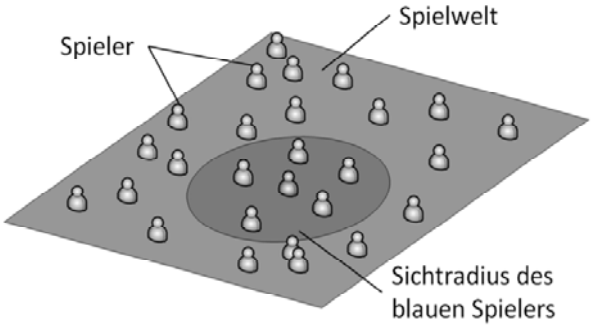



**Abbildung 1: Spielwelt eines MMOG**

Das erste Problem, das ins Auge fällt, ist die enorme Anzahl von Spielern. Sind es bei Multiplayer Games zwischen 2 und bis zu etwa 50 Spielern, die sich an einer Spielrunde beteiligen, können es bei MMOGs im Normalfall mehrere Tausend Spieler sein. In Extremfällen, wie z.B. bei EVE-Online, sind es bis zu über 50.000 Spieler[6], die geleichzeitig an einer einzigen Spielrunde teilnehmen. Für diese riesige Anzahl von Spielern müssen Möglichkeiten zum effizienten Datenaustausch zur Verfügung gestellt werden. Da die Kommunikation in der Regel über Internet geschieht, muss auch dementsprechend Bandbreite vorhanden sein. Des Weiteren wird Hardware benötigt welche die Kommunikation koordiniert, damit jeder Spieler die Daten bekommt, die er benötigt.

Das zweite große Problem ist das Spiel bzw. der Aufbau des Spiels selbst. In der Regel hat man es mit einer 3-dimensionalen virtuellen Welt zu tun, in der sich alle Spieler bewegen. Das bedeutet zunächst dass Hardware zur Verfügung stehen muss, auf der die Welt simuliert und berechnet werden kann. Da sich die Welt und die Spieler ständig ändern, müssen auch Möglichkeiten der Speicherung gegeben sein. Ein weiteres Problem, welches sich aus dem Aufbau ergibt ist die Konsistenz: Jeder Spieler sollte die gleichen Gegebenheiten vorfinden.

## 3. Klassische Lösungen
### 3.1 Client-Server-Modell
Beim klassischen Client-Server-Modell steht ein zentraler Server zur Verfügung, welcher die gesamte Spielwelt und Kommunikation verwaltet. Die einzelnen Spieler stellen jeweils die Clients

dar. Jeder Client bekommt nur die Informationen vom Server die er benötigt. In der Regel sind das Daten über Objekte aus der näheren Umgebung - meist dem Sichtradius der gesteuerten Spielfigur. Jede Statusänderung eines Clients, wie z.B. das Bewegen einer Spielfigur wird vom Client direkt an den Server gesendet und erst dieser leitet sie an andere Clients weiter.

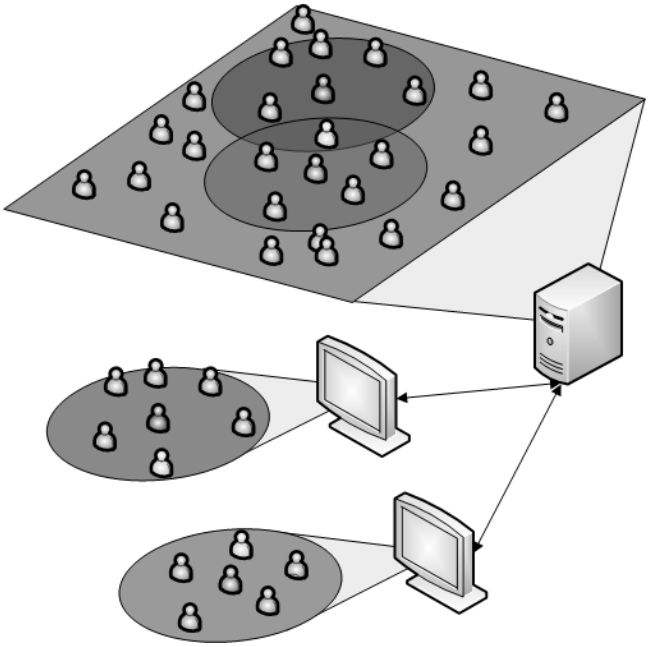


**Abbildung 2: Klassisches Client/Server-Modell**

Ein Vorteil dieses Systems ist, dass durch die ausschließliche Kommunikation über den Server und der Berechnung der Spielwelt auf diesem der Server „allwissend“ ist. Dadurch können Manipulationen wie Cheating relativ leicht erkannt und verhindert werden. Des Weiteren können so leicht die aktuellen Zustände der Spielwelt abgespeichert werden. Das Konsistenzproblem wird soweit gelöst, indem angenommen wird, dass der Server die richtige Spielwelt vorgibt und nur die Spieler über den aktuellen Zustand informiert werden müssen. Ein weiterer Vorteil dieses Modells ist, dass keine Hardware beim Client für die Simulation der Spielwelt belastet wird, er muss nur die Grafikausgabe noch berechnen. Außerdem ist dieses Modell leichter zu realisieren, da es relativ unkompliziert ist und schon ausreichend Implementierungen vorhanden sind.

Auf der anderen Seite bringt das klassische Client-Server-Modell auch eine ganze Reihe Nachteile mit sich. Zum einen ist die Simulation der Spielwelt auf dem Server verhältnismäßig teuer, da die Hardware dementsprechend vorhanden und dimensioniert sein muss. Zum Anderen werden sehr viele Verbindungen von verschiedenen Clients zum Server aufgebaut und durch die ausschließliche Kommunikation über den Server entsteht ein extrem großes Datenaufkommen, was demzufolge eine extrem gute Anbindung ans Internet mit genügend Bandbreite erfordert. Außerdem benötigen Nachrichten an andere Clients zwingend immer 2 Hops, wobei hier ein Hop den Weg von einem Rechner zum nächsten darstellt.[3]

## 3.2 Distributed-Server-System

Das Distributed-Server-System unterscheidet sich vom normalen Client-Servermodell dadurch, dass auf Server-Seite nicht ein einzelner Rechner vorhanden ist, sondern die Arbeit auf mehrere Server verteilt wird. Wie die Verteilung aussieht ist stark vom Aufbau des jeweiligen Spiels abhängig. Meist wird aber die Spielwelt in verschiedene Teile aufgeteilt und für jeden Teil steht dann ein extra Server zur Verwaltung und Simulation bereit. Auch können Sonderaufgaben, wie z.B. Login oder Ingame-Kommunikation auf extra Server ausgelagert sein. Für den Client stellen sich die Server aber als ein einziger virtueller Server dar.

Durch die Verteilung der Aufgaben auf mehre Server wird eine ausgeglichene Belastung der einzelne Server erreicht. Außerdem skaliert das Spiel bei steigender Spieleranzahl besser.

Die Verteilung wirkt sich aber nachteilig auf das Konsistenzproblem aus: Die einzelnen Spieler und Objekte bewegen sich zwischen den Teilen und damit auch zwischen den Servern hin und her. Jedes Objekt muss beim Wechseln des Servers kopiert werden. Dabei darf es danach nicht auf beiden Servern vorhanden sein. Genauso muss gewährleistet sein, dass es vollständig kopiert wurde. Dazu kommt noch, dass die Codekomplexität steigt, da jetzt nicht nur das Spiel selbst, sondern auch noch die Verteilung auf die einzelnen Server berechnet werden muss. Und auch die Lösungen um das Konsistenzproblem in den Griff zu bekommen, wirken sich negativ auf die Codekomplexität aus.

Ansonsten gelten hier auch die gleichen Vor- und Nachteile des normalen Client-Server-Systems.[3]

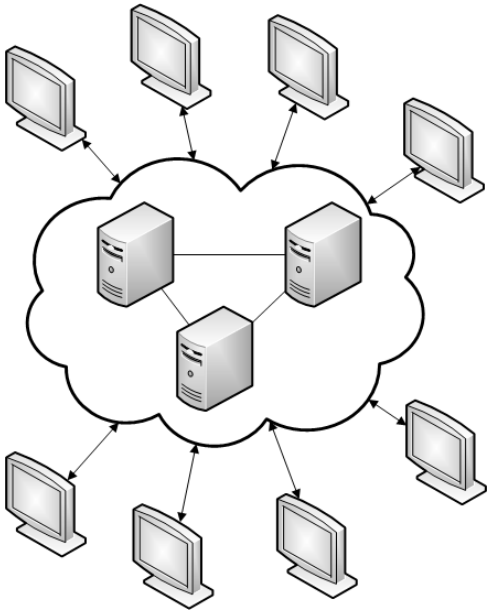


**Abbildung 3: Distributed-Server-System**

## 3.3 Bewertung der Klassischen Systeme

Bei beiden Systemen wird die Zentralisierung zum Flaschenhals: Sowohl beim normalen Client-Server-System, als auch beim Distributed-Server-System hängt die Qualität des Spiels von der den Servern zur Verfügung stehenden Internetanbindung und der Hardwaredimensionierung ab. Im April 2009 zeigte sich dieses Problem deutlich bei dem Multiplayer-Echtzeit-Strategispiels Demigod: Der Anbieter des Spiels hatte für die Anfangszeit die Server und Internetanbindung auf die Anzahl der produzierten und in den Handel gebrachten Kopien ausgelegt. Da für die ersten Wochen etwa 18.000 Spieler erwartete wurden, hätte die Leistung der Server, welche 50.000 Spieler gleichzeitig verkraftet hätten, eigentlich reichen müssen. Nur hatte der Anbieter die Raubkopierer vergessen: Obwohl das Spiel durch Keys gesichert ist, konnten sich über 120.000 Spieler auf die Server einloggen, was natürlich zum Zusammenbruch führte, bzw. das Spiel unspielbar machte. Dieser Umstand sorgte dafür, dass die Fachpresse das Spiel mit etwa 60-70% der möglichen Punkte bewertet, was sich wiederum negativ auf die Verkaufszahlen auswirkte und somit negative wirtschaftliche Folgen hatte.[8]

Ein zweiter Knackpunkt der klassischen Systeme: Die Server werden zum Single point of failure. Fällt ein Server aus, kann das ganze Netzwerk bzw. das Spiel zusammenbrechen. Vor allem wenn der eine oder bei verteilten Systemen ein wichtiger Server, wie z.B. der Login-Server ausfallen können alle Spieler nicht mehr oder nur noch bedingt am Spiel teilnehmen.

# 4. Allgemeiner Lösungsansatz von P2P

Der allgemeine Ansatz zur Lösung der Probleme von klassischen Systemen und der neuen Probleme, die sich aus P2P ergeben sieht folgender maßen aus: Zunächst sollen Spieler Informationen aus ihrem Sichtradius, die sie benötigen möglichst schnell bekommen. Vor allem die Reaktion auf sich im Sichtradius bewegende Spieler sollte möglichst schnell sein, sprich der Datenaustausch sollte über wenige HOPs gehen. Andersherum sollten Spieler, die außerhalb des Sichtradius sind kaum unnötige Information über die eigene Spielfigur bekommen und falls sie dennoch Nachrichten benötigen, reicht es wenn sie sie über mehre HOPs erreichen. Insgesamt soll eine gute Skalierbarkeit mit sehr vielen Spielern erreicht werden.

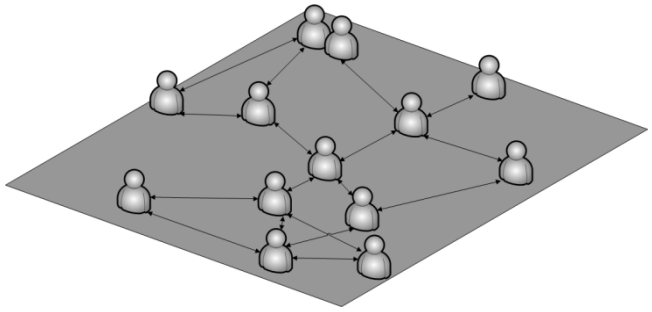


**Abbildung 4: P2P-Ansatz in der Spielwelt**

Im Allgemeinen werden diese Ziele auf folgende Weise erreicht: Die einzelnen Spieler stellen im Netzwerk die einzelnen Peers dar. Das heißt alle Spieler sind gleichberechtigt mit ein paar anderen Spielern direkt verbunden. Informationen werden an alle Spieler, mit denen eine direkte Verbindung besteht gesendet und entsprechend weitergeleitet. Am besten sollten Spieler, die sich in der Spielwelt in der gleichen Umgebung befinden möglichst über wenige andere Nodes im P2P-Netz verbunden sein, damit die Nachrichtenübertragung in wenigen Hops realisierbar ist. [3]**Fehler! Verweisquelle konnte nicht gefunden werden.**

# 5. Beispiele für P2P-Spiele-Lösungen

## 5.1 SimMUD, ein reine P2P-Lösung

### *5.1.1 Netzwerkaufbau*

SimMud baut sein P2P-Netz mit Hilfe des Overlay PASTRY auf. PASTRY wiederum benutzt zur Verwaltung Distributed Hash Tables und organisiert die einzelnen Peers in einer Ringstruktur. Den einzelnen Daten werden über eine globale Hashfunktion Objekt-Schlüssel zu eindeutigen Knoten zugewiesen. Dadurch wird eine Binärsuche nach Daten möglich, welche in O(log(n)) liegt. Außerdem stellt PASTRY schon Implementierungen für das Entfernen und Hinzufügen von neuen Knoten, als auch für die Fehlerbehandlung, im Falle von Ausfällen einzelner Knoten, bereit.

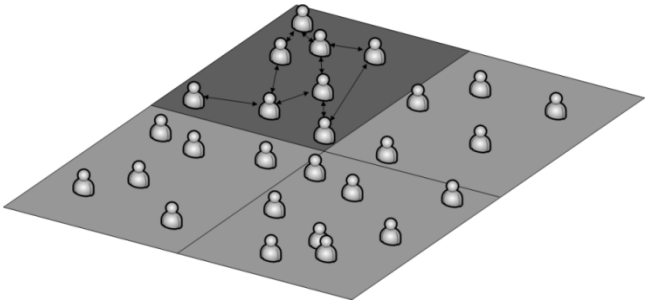


**Abbildung 5: Übersicht SimMUD**

Für die Informationsverbreitung benutzt SimMUD SCRIBE, welches ein Publisher/Subscriber-System darstellt, dass auf PASTRY aufbaut. Nur die Peers, die sich bei einem bestimmten Publisher eingetragen haben bekommen auch Daten von diesem. Anders als bei Broadcasting-Systemen wird so das Fluten des Netzwerkes verhindert und gleichzeitig eine bessere Skalierung mit vielen Nutzern erreicht. [1][9]

### *5.1.2 Spielaufbau*

Die Spielwelt ist in SimMUD wie in Abbildung 5 zu sehen ist, in feste Regionen mit fester Größe eingeteilt. Für jedes Objekt in der Spielwelt gibt es einen „Coordinator“. Auch für die Regionen, welche auch nur Objekte darstellten, gibt es Coordinators. Jeder Coordinator stellt dabei den Publisher für das Objekt dar. Jeder Peer stellt mehre Coordinators für verschieden Objekte bereit und gleichzeitig auch noch den für die eigene Spielfigur. Außer auf den einzelnen Coordinatoren werden Objektdaten auch auf anderen Peers als Sicherheitskopie gespeichert. Bei einem Ausfall kann so ein anderer Peer als Ersatz einspringen.

Die einzelnen Peers innerhalb einer Region bilden eine Gruppe, in der alle Peers nicht unbedingt direkt miteinander vernetzt sind. Informationsaustausch findet über das Publisher-Subscriber-System statt, was bedeutet dass Peers nur Information von Coordinators erhalten, bei denen sie sich als Subscriber angemeldet haben. Für Interaktion zwischen zwei Spielern wird eine direkte Verbindung aufgebaut. Will ein Spieler ein Objekt verändern, sendet er diese Information an dessen Coordinator. (nach[1][3])

### *5.1.3 Performanz / Bewertung von SimMUD*

Mit dem oben beschrieben Aufbau ist es möglich, benötigte Nachrichten innerhalb von durchschnittlich unter 6 Hops an den Client zu liefern, der diese benötigt. Je nach Latenzzeit sind somit Reaktionsgeschwindigkeiten von bis zu unter 200ms möglich. Leider geht das Verfahren überhaupt nicht auf das Thema Cheating ein, im Gegenteil, jeder Coordinator kann die Objekte, die er verwaltet selbst manipulieren. Insbesondere ist es für den Client möglich die eigene Spielfigur zu verändern, ohne dass es andere merken. (nach[3])

## 5.2 FreeMMG, eine Hybrid-Lösung

### *5.2.1 Allgemeines zu FreeMMG*

FreeMMG baut ein Hybrid-Netz aus P2P und einem Server/Servercluster auf. Der Server ermöglicht den Login in das Spiel und stellt somit auch den Einstiegspunkt für das P2P-Netz dar. Zusätzlich übernimmt der Sever das Session-Tracking und stellt Möglichkeiten bereit, die Spielwelt abzuspeichern. Zuletzt übernimmt der Server die Verwaltung der „Areas of Interest“ und damit auch die Verteilung der Clients auf diese.

Die Einzelnen Peers in FreeMMG übernehmen die Simulation der Spielwelt. Dazu haben alle Peers in einer Area of Interest eine Kopie dieser und tauschen vorher die Aktionen aus die ausgeführt

werden sollen. Anschließend berechnen alle den nächsten Zustand. (nach[2][3])

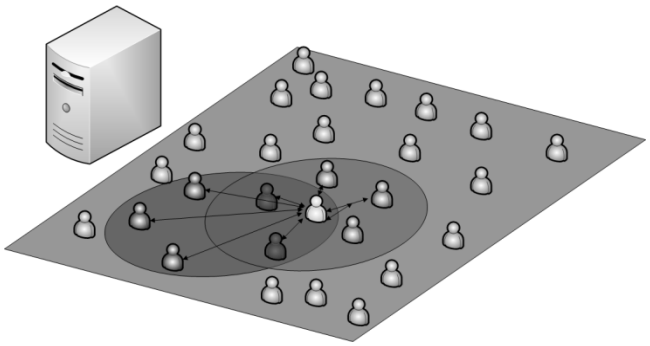


**Abbildung 6: Übersicht FreeMMG**

### *5.2.2 Spielaufbau*

„Areas of Interest“ stellen die Interessensgebiete der Clients dar. Sie sind nicht nur eine Region der Spielwelt, in der sich die Spielfigur des Clients befindet, sondern können auch ein bestimmter Chat-Kanal, eine Art Marktplatz/Auktionshaus oder Sonstiges sein. Aus diesem Grund können Clients auch an mehreren Areas of Interest teilnehmen. Innerhalb eines Interessengebietes werden zwischen den einzelnen Clients Direktverbindungen aufgebaut. Aktionen zwischen Objekten sind nur möglich, falls sich beide Objekte in der gleichen Area of Interest befinden. Abbildung 6 zeigt einen möglichen Aufbau. Der Peer in der Mitte hat hier 2 Areas of Interest, einmal seinen Sichtradius (Kreis um die Spielfigur) und eine zweite z.B. ein Gruppen-Chat-Kanal mit den Spielern links von ihm. Mit jedem Peer in den beiden Areas besteht eine Direktverbindung. (nach[2][3])

### *5.2.3 Cheating in FreeMMG*

Cheating wird in FreeMMG zumindest stark erschwert. Clients führen innerhalb einer Area die gleichen Berechnungen auf dem gleichen Datenbestand durch. Würden Cheater unter den Clients sein, würden ihr Ergebnisse von denen der Mehrheit abweichen und die Ergebnisse könnten als falsch betrachtet werden.

Da der Server die Verwaltung der „Areas of Interest“ übernimmt, können die Clients die Verteilung nicht beeinflussen. Bei FreeMMG werden die einzelnen Clients durch den Server zufällig in verschieden Areas gesteckt. Das hat zur Folge, dass es auch Gruppen von Cheatern erschwert wird, ihre manipulierte Version des Spielzustandes als den gültigen Zustand durchzusetzen. (nach[2][3])

### *5.2.4 Bewertung von FreeMMG*

Durch die direkte Verbindung innerhalb einer Area of Interest können Daten an die Clients, die sie benötigen innerhalb eines Hops geliefert werden. Auch der Versuch Cheating zu unterbinden fällt positiv aus.

Nur leider benötigt FreeMMG immer noch einen Server und dieser wird, wie bei den Klassischen Systemen wieder zum Flaschenhals. Versuche haben gezeigt, dass bei FreeMMG der Traffic, der über den Server geht ab ca. 300 Clients nicht mehr linear skaliert. Deshalb lassen sich kaum Aussagen darüber machen, ob sich die Implementierung für ein typisches MMOG mit mehreren Tausend Spielern eignet. Auch die Cheat-Sicherheit ist nur bedingt gegeben, da die Verteilung nur zufällig ist, und deshalb auch zufällig mehre Cheater wieder in eine Area kommen könnten. (nach[2][3])

## 5.3 pSense

### *5.3.1 Algemeines zu pSense*

pSense setzt im Wesentlichen wieder wie SimMUD auf ein reines P2P-Netz. Die einzelnen Clients stellen mit ihrer Spielfigur die Peers im Netz dar. Leider macht das Paper pSense keine genaueren Angaben wie die Spielwelt simuliert wird, sondern bezieht sich eher auf einen Algorithmus, welcher Figuren verwaltet die sich in einer Spielwelt bewegen. (nach[4])

### *5.3.2 Near-Nodes*

Zunächst werden in pSense so genannte „Near-Nodes“ definiert. Near-Nodes sind Peers, deren Spielfigur sich innerhalb des Sichtradius des lokalen Spielers befindet (siehe Abbildung 7). Diese Near-Nodes, bzw. die entsprechende Peers werden beim lokalen Peer in Listen organisiert. Zu diesen Near-Nodes besteht immer eine Direktverbindung. Updates über den eignen Zustand, werden an alle Near-Nodes gesendet. Dadurch können Informationen innerhalb eines Hops mit allen Peers, die die Informationen möglichst sofort benötigen ausgetauscht werden. Nodes, die den Sichtradius verlassen werden einfach aus der Liste entfernt und bekommen in der Zukunft keine Information mehr, was wiederum den Traffic gering hält. Peers, die in den Sichtradius eintreten werden über so genannte Sensor-Nodes erkannt. (nach[4])

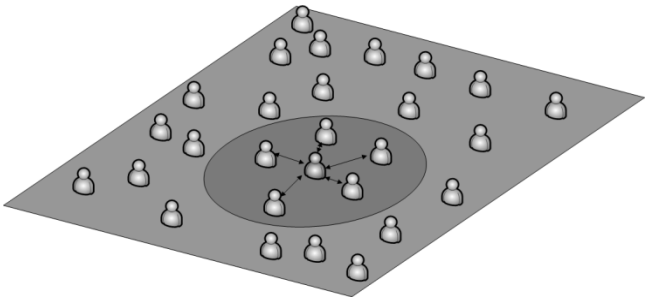


**Abbildung 7: Near-Nodes (innerhalb des Kreises)**

### *5.3.3 Sensor-Nodes*

„Sensor-Nodes“ stellen die Peers dar, welche in bestimmten Richtungen/Winkeln die nächsten Peers außerhalb des Sichtradius sind (dunkel grauen Figuren in Abbildung 8). Diese bleiben dem lokalen Peer weiterhin bekannt. Alle Updates des lokalen Peers werden auch an seine Sensor-Nodes weitergeleitet. Falls ein noch unbekannter Peer in den Sichtradius eintritt, ist er einem Sensor-Node schon bekannt und dieser kann es dem lokalen Peer mitteilen. Zwischen dem Sensor-Node und dem neuen Peer besteht vorher schon eine Near-Node-Beziehung. Auch kann der Sensor-Node entscheiden, ob er selbst noch die Kriterien eines Sensor-Nodes erfüllt, oder ob es einen anderen Peer außerhalb des Sichtradius gibt, der besser geeignet wäre. In diesem Fall würde der lokale Peer darüber informiert werden und seine Listen dementsprechend umschreiben. (nach[4])

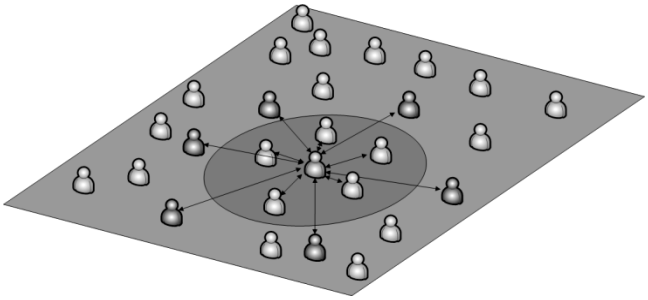


**Abbildung 8: Sensor-Nodes (dunkelgrau, außerhalb des Kreises)**

### *5.3.4 Bewertung von pSense*

Auch hier wird durch die direkte Verbindung innerhalb des Sichtradius eine maximale Übertragungszeit von einem Hop gewährleistet. Auch regelt der Algorithmus genau wie mit sich bewegenden Spielern/Objekten umgegangen werden soll.

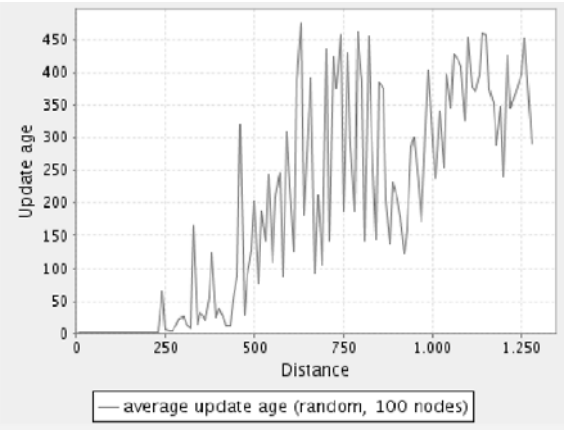



**Abbildung 9: Dauer der Nachrichtenübertragung (aus[4])**

Für Nachrichten, die für Peers außerhalb des Sichtradius bestimmt sind, können die Laufzeiten kaum mehr vorhergesagt werden, da es keinen Algorithmus zur effektiven Weiterleitung gibt. Leider sind dazu eine Menge weitere Probleme ungelöst. Zum Beispiel fehlen Lösungen zum Verwalten von Objekten oder der Spielwelt an sich. Genauso werden Login oder Verlassen des Spieles nicht beachtet. Cheating wird bei dieser Lösung auch nicht angesprochen. **Fehler! Verweisquelle konnte nicht gefunden werden.**

## 6. Allgemeine Bewertung der P2P-Lösungen

Insgesamt lässt sich behaupten, dass serverunabhängige Lösungen für MMOGs mit der Hilfe von P2P realisierbar wären, nur haben alle Vorgestellten Lösungen noch zu viele Probleme.

Alle Verfahren sind mehr oder weniger noch theoretischer Natur. So bieten alle Lösungen nur Antworten auf Teilprobleme und lassen wichtige Aspekte außen vor. So gehen SimMUD und pSense nicht auf das Thema Cheating ein und bieten hier keine Lösung. FreeMMG benötigt immer noch einen Server und ist deshalb kaum besser als die klassischen Systeme. pSense bietet eigentlich nur eine sehr genaue Lösung für ein Teilproblem.

Dazu kommt noch, dass auch die Performanzanalysen nicht 100% mit der realen Welt übereinstimmen. Sie wurden alle unter „Laborbedingungen“ gemacht. Über das Verhalten der Systeme im realen Internet ist wenig bekannt.

Und schließlich werden bei allen Lösungen allgemeine Probleme von P2P nicht behandelt. Alle Lösungen setzten voraus dass ein P2P-Netz mehr oder weniger schon besteht. Wie dieses im Internet dauerhaft aufrechterhalten bleiben soll wird nicht besprochen. Auch für die Sicherheitsrisiken die P2P mit sich bringt werden keine Lösungsvorschläge gemacht. Zum Beispiel müssen die IP-Adressen der einzelnen Clients jedem anderen Client bekannt sein, was neue Angriffspunkte mit sich bringt und ein „anonymes“ Spielen nicht erlaubt. Zusätzlich könnte über ein Spiel auch schadhafter Code auf die Clients verteilt werden.

## 7. Einsatz von P2P in heutigen Spielen

Trotz der im vorherigen Abschnitt genannten Probleme, wird P2P heute schon in Spielen verwendet. So nutzt Blizzard P2P, um die Patches für „World of Warcraft“ zu verteilen und somit die eigenen Download-Server zu schonen. Andere auf dem Markt befindliche Spiele, die P2P in oben beschriebener Weise nutzen sind mir nicht bekannt. Auch EVE-Online, welches in regelmäßigen Abständen neue Rekorde mit gleichzeitig in ein einziges Spiel eingeloggten Speilern bricht, vertraut einzig und allein auf einen riesigen Servercluster, welcher die gesamte Spielwelt berechnet.

Der „Hauptnutzen“ von P2P in Verbindung mit Spielen, ist leider immer noch das Verteilen von Raubkopien.

## 8. Zusammenfassungen und Ausblick

Durch den Einsatz von P2P-Lösungen könnten man Serverunabhängige MMOGs realisieren. Diese Spiele könnten auch mit einer großen Anzahl von Nutzern noch gut skalieren. Außerdem würden sie Hardware und Internetbandbreite auf Anbieterseite sparen helfen.

Bis es dazu kommt, ist es aber noch ein langer Weg. Es müssen zuvor noch cheatsichere Lösungen gefunden werden. Dazu kommt noch, dass auch Login und Logout oder Ausfälle von Peers behandelt werden. Und zuletzt müssen noch die allgemeinen Probleme von P2P gelöst werden.

Daraus ergibt sich, dass die Aussichten relativ ungewiss sind. Zum einen haben die Klassischen Systeme immer noch ihren Vorteil, sonst würden sie nicht so erfolgreich eingesetzte werden. Zum Anderen gehen neue Entwicklungen den entgegengesetzten Weg: Da die Hardware der meisten Spieler im Moment relativ vielfältig ist und die Internetbandbreite mittlerweile für Videoübertragung auch in hohen Auflösungen gut genug ist, überlegen Anbieter auch noch die Grafikberechnung auf Servern auszuführen und nur noch das Video durchs Internet zu senden.

## 9. Literatur


[1] Björn Knutsson, Honghui Lu, Wei Xu, Bryan Hopkins, “Peer-to-Peer Support for Massively Multiplayer Games”, in Proceedings of IEEE INFOCOM 2004, The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, 2004

[2] Florian Dautermann, „Implementierung und Evaluierung einer manipulationsresistenten Peer-to-Peer Architektur für Multiplayer Online Games“, Diploma-Thesis, TU Darmstadt, 2007

[3] Markus Sinner, “Massive Multiplayer Games a Peer to Peer Framework”, Diploma-Thesis, TU Darmstadt, 2005

[4] Arne Schmieg, Michael Stieler, Sebastian Jeckel, Patric Kabus, Bettina Kemme, Alejandro Buchmann, "pSense - Maintaining a Dynamic Localized Peer-to-Peer Structure for Position Based Multicast in Games", in Proceedings of the 2008 Eighth International Conference on Peer-to- Peer Computing, 2008

[5] Blizzard Entertainment - Press Release http://eu.blizzard.com/de/press/081223.html

[6] www.areagames.de: News - EVE-Online http://www.areagames.de/artikel/detail/EVE-Online-Mehr-als-300-000-aktive-User/101086

[7] 4players.de: Wartezeiten in Tausendwinter (WoW) http://wowsource.4players.de/news,2482,Wartezeiten_in_Ta Tausendwint.html

[8] 4players.de: Demigod: Überlastung durch Raubkopierer http://www.4players.de/4players.php/spielinfonews/Allgemei A/10163/1887648.html

[9] Christian Webel, „Pastry und SCRIBE”, TU Kaiserslautern, http://vs.informatik.uni-kl.de/lehre/SeminarWiSe04-05/Vortraege/Termin5/Thema8.pdf

# The Application of Common Peer-to-Peer Concepts in the Field of Distributed Databases

Marius Treitz
Betreuer: Bernhard Amann
Seminar Innovative Internettechnologien und Mobilkommunikation SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: treitz@in.tum.de

***Abstract*— Peer-to-Peer technology has been a successful concept in many fields of application. This paper explores how the Peer-to-Peer paradigms can be applied to databases. In detail, this paper discusses the basic concepts of Peer-to-Peer technology, the similar concepts in relation to Peer-to-Peer databases and the key issues of such systems. It further discusses the benefits and potential (future) applications of such a system. Throughout the paper, two Peer-to-Peer database projects are continuously mentioned – the XPeer project and the Piazza Peer Data Management System.**

***Keywords*— Peer-to-Peer databases, Peer-to-Peer networks, Distributed Databases, P2P, Peer-to-Peer**

## I. Introduction

The last years have seen an increased use of peer-to-peer concepts in many areas of application; widely portrayed is the example of filesharing [1]. Typical characteristics of such systems include the „shared provision of distributed resources and services, decentralization and autonomy“ [2].

The success of these concepts and principles has lead experts to attempt to apply these concepts to the field of databases, hoping to take advantage of some of the benefits of peer-to-peer systems over their classic distributed counterparts sharing drawbacks such as their „static topology“ and „heavy administration work“ [3].

While certain large databases, such as the Google BigTable project [4], and some academic projects already make use of peer to peer concepts in the field of databases, there are key open challenges to be addressed.

The paper begins by outlining some of the principles of peer-to-peer networks; these are referenced throughout the paper and in general apply to peer-to-peer databases as they apply to all peer-to-peer systems. It goes on to describe how some of the main concepts of peer-to-peer networks relate to peer-to-peer databases; the benefits and challenges are explained in detail. Furthermore, some incentives for the employment of peer-to-peer databases are explained. At last, the paper introduces some potential applications of peer-to-peer databases and provides some real-world examples of possible uses of the technology.

## II. Principles of peer-to-peer Technology

In recent years, the principles of peer-to-peer technology have been widely discussed for applications such as filesharing via Peer-to-Peer platforms [1]. The following section provides a short overview of design principles and typical characteristics of peer-to-peer systems. It also outlines some of the key economic incentives that drive the use of such systems. This section will be the basis for outlining Peer-to-Peer databases in section IV, as the fundamental principles of Peer-to-Peer technology hold true for Peer-to-Peer databases as well. Peer-to-Peer filesharing is used as a running example, as it could be perceived as a basic version of peer-to-peer databases.

### A. Key Concepts

Schoder & Fischbach provide a basic definition of the term Peer-to-Peer:

> P2P refers to technology that enables two or more peers to collaborate spontaneously in a network of equals (peers) by using appropriate information and communication systems without the necessity for central coordination [1].

In addition to this statement, three more basic characteristics of Peer-to-Peer networks should be mentioned: the „shared provision of distributed resources and services, decentralization and autonomy“ [2]. Some basic examples of peer-to-peer technology are the fields of „content sharing“, „storage services“ or – in the field of services – „distributed 'grid' computation“ [5].

### B. File Sharing Peer-to-Peer Networks

The concept of Peer-to-Peer networks can be applied to numerous applications, one such application is the sharing of files between the nodes of a Peer-to-Peer network. In this use case, „peers that have downloaded the files in the role of a client subsequently make them available to other peers in the role of a server“ [2]. Examples of such file sharing networks are Napster, Freenet or Gnutella [6]. The concept of a Peer-to-Peer file sharing network bears similarities to the concept of a peer-to-peer database and will later serve as a reference when discussing Peer-to-Peer databases.

### C. Hybrid Peer-to-Peer Networks

Some networks make use of certain Peer-to-Peer concepts, yet do not adhere to all of them; such networks can be referred to hybrid Peer-to-Peer networks [2]. In a pure Peer-to-Peer network, all nodes are considered truly equal, and there is no central governing instance in the network [6]; In a hybrid Peer-to-Peer network however, there is a two-step interaction: A client first communicates with a central instance to exchange specific metadata, which is then used for communication with other peers in the network [6]. Another concept that defies the principle of equality between all peers is the concept of superpeers, where a group of peers perform „selected functions, such as indexing or authentication“ [2]; these can be perceived as an intermediate step between one or more fully centralized servers and fully autonomous and equal clients [6].

Milojicic et al. provides a scale from „pure“ to „hybrid“ networks, ordered by their „degree of centralization“; these are (from pure to hybrid): pure node equality, the superpeer model, multiple dedicated servers and finally, a single dedicated server.

*1) Advantages of Pure Peer-to-Peer Networks:* An example of an application using a pure peer-to-peer model is the Gnutella network; it makes use of a „flooded request model“ to query for files made available by other peers in the network, which must be limited in its spreading across the network to avoid unwanted proliferation; the advantage of such a decentralized architecture is „improved scalability“ [6] - the network may grow without the addition of special peers that would handle the additional load.

*2) Advantages of Hybrid Peer-to-Peer Networks:* The concept of multiple dedicated servers employed for administrative services is made use of in the Napster network which in turn allows for a „centralized directory model“ for file searches; central instances however are a threat to scalability, as the ability to serve requests must rise with increasing demand from the equal peers in the network [6].

For the latter part of the paper, when considering Peer-to-Peer databases, only pure Peer-to-Peer networks will be considered for discussion; several distributed databases area already in use, such as the Google BigTable of which may be said that it has some peer-to-peer characteristics, such as the dynamic addition of so called „Tablet Servers“ to „accomodate changes in workloads“ [4].

### D. Finding Data in Peer-to-Peer Filesharing Networks

This section briefly explains two methods of locating files in a hybrid, as well as in a pure peer-to-peer network.

Locating files in the network can be performed using a mapping from keys – such as a file name – to the actual location of the file in the network [7]. Therefore, queries in a Peer-to-Peer network are usually „requests for objects by identifier“ rather than more complex queries as employed in Peer-to-Peer databases [5]. This search behavior can be realized using a „centralized directory model“, employed in the hybrid Peer-to-Peer file sharing network Napster, in which a central instance possesses an overview of the network and is contacted first by peers to obtain information about the location of a file in the network [6]. An opposing, decentralized approach is applied with the „flooded request model“, implemented in the file sharing network Gnutella; in this model, a peer floods the network with its query, starting with its own peers, and limiting the request by defining a „time to live“ for the request [6].

### E. Economics of Peer-to-Peer Networks

Peer-to-peer technology may provide various benefits. [2] lists „improved scalability, lower cost of ownership, self-organized and decentralized coordination of previously unused or limited resources, greater fault tolerance, and better support for building ad hoc networks.“ It is notable that this includes not only economic benefits such as greater fault tolerance, which may otherwise be expensive to achieve in classic distributed systems; the list also mentions advantages in usability, such as the ability to build „ad hoc networks“ – a usability advantage demonstrated by filesharing networks in which it is easily possible to enter an existing network for clients [2].

### F. Limitations

As previously mentioned, the concept of peer-to-peer filesharing is closely related to the concept of sharing semantically rich data as it be the case in a Peer-to-Peer database. Several key drawbacks of the Peer-to-Peer filesharing paradigm however prevent a simple extension of this already existing and proven concept. This is foremost the „lack of semantics“, the limitation to „large-granularity requests“ and the search method of requesting „objects by identifier“ [5].

The lack of semantics is an obvious fault. Databases are usually based on the storage of complex data – such as tree-based data forests in the case of XML or relational data – that can be queried using languages such as XPath (XML) or SQL. The shared objects in Peer-to-Peer filesharing networks are usually whole files and they are only transferred in their state as entire files; it is not possible to alter a file by means of a query definition, as it would be possible in a database, not to mention more complex tasks such as joins, a concept supported by relational databases that allows for dynamic consolidation of various data sets. Gribble et al. mention this as another conceptual problem [5]. Peer-to-Peer filesharing networks are usually based on the mapping of filenames to the respective location of the files in the network [7]. This method of querying is entirely unrelated to the more complex concept of query formulation employed in the field of databases.

The following section details some of the concepts shared by P2P databases and P2P file sharing, and especially details some of the special features and issues in the field of P2P databases.

## III. Concepts of P2P Databases

The previous section of the paper outlined the basic technical architecture of P2P networks and how this architecture can be utilized to create simple data sharing networks by the example of file sharing. This section demonstrates how

P2P concepts can be employed for the use of so called P2P databases, or „peer data management system“ – short „PDMS“ [8]. At first, an overview of the fundamental principles is provided, followed by an in detail description of some of the concepts.

At the time of this paper, the author was not able to find a significant number or great variety of implemented Peer-to-Peer databases. The following description is therefore based mainly on the two Peer-to-Peer database networks XPeer and the Piazza PDMS.

### A. Single Node Autonomy

The fact that single nodes in a peer-to-peer network possess autonomy is one of the fundamental concepts of peer-to-peer networking [2]. In this context, Sartiani et al. bring up this issue of a „changing topology“ in a peer-to-peer database network, arguing that „peers are autonomous, in the sense that they are free to choose the data to contribute to the system, to manage local data without external constraints, and to connect and disconnect at any time“ [3].

In a private P2P network, in which all the single physical nodes are controlled by a single or multiple trusted entities, it may be possible to local rules to limit the autonomy of individual nodes or prevent scenarios in which single nodes act as autonomous entities against the best interest of the network as a whole; in a true P2P network however, it may become difficult to impossible and – depending on the use case – undesirable to limit each node's ability to create, delete or alter local database content or even join or leave a specific peer-to-peer database network [3].

### B. Schema Mappings

In traditional, stand-alone databases, such as local, relational databases, one usually defines a detailed data schema before data entry and subsequent selection and modification is possible. As an example, prior to entering data into a MySQL database, a table must first be created, and the table's acceptable input must be clearly defined. As a result, the table, enriched with data, no matter how large, will always contain a homogeneous set of tuples.

In peer-to-peer databases, this poses a problem, as not all available data on all nodes in the network may be homogeneous; in fact, Piazza, a „peer data management system“ was designed on the principle of a „system that enables sharing heterogeneous data in a distributed and scalable way“ [8], [9]. As a solution to this problem, Tatarinov et. al use schema mappings in the Piazza system.

A schema mapping is a contextual mapping connecting different data schemas between the peers. Independant peers in a network, each possessing a set of schemas with underlying data, could be connected with each other, by mapping their respective schemas. From a global network perspective, this would lead to a tightly woven net of heterogeneous schemas, all of which are interconnected through schema mappings. A query in the network would be executed by a peer, based on its own available data schemas, but could relate to data on other peers, by expanding the query by including foreign schemas through said schema mappings [9].

In the Piazza project, the mapping effort is twofold: so called „storage descriptions“ map the stored data of an individual peer to a „peer schema“, while „peer descriptions“ create the connection between the „peer schemas“ at various nodes of the network [9].

### C. Querying

The querying of data in a peer-to-peer database system is a complex task; complete queries throughout an entire peer-to-peer data system are difficult, as was already shown with the more basic example of atomic data in P2P file sharing in section II-D. The key issues in this respect are especially the formulation of queries, the efficient iteration of the network for available information and the optimization of such queries.

*1) Formulation of Queries:* Query formulation in a stand-alone relational database is limited in its complexity. As mentioned previously, if all data in a single, local database is clearly defined by schemas, a query matched against these schemas can produce clear, definite and complete results. Peer-to-peer databases however may not necessarily be able to provide an absolute definition of a global, homogeneous schema – this would likely require a hybrid system with a central authority, as data is heterogeneous and a query may require data that is not available on one single node, but distributed over a multitude of nodes.

The two systems XPeer and Piazza offer solutions to resolve the issue of query resolution. Both rely on mapping the heterogeneous schemas of nodes to produce a result set, they differ however in their method of query execution. In both projects, data is stored in the form of XML based forests on the nodes of the network; the data on each node is described with what from now on will be commonly referred to as local schemas; a schema connection between nodes is established by mapping the local schemas on multiple nodes [3], [9].

XPeer makes use of the FLWR query language, a language developed to query XML trees; the FLWR queries are transformed „into algebraic expressions“ and can then be processed into a complete query using a rule set; this formulated query is then forwarded to the parent in the XPeer hierarchy to begin the traversal and data finding algorithm. [3].

The Piazza PDMS makes use of a similar query resolution algorithm by resolving individual sub parts of the original query into atomic bits that reference the stored locations of the mappings only; a query on the network is based on a the schema of the querying node and additional data from other nodes may be utilized to complete the query by the use of existing peer mappings [9].

*2) Network Iteration:* A main advantage of Peer-to-Peer database networks, as well as one of its main challenges, is the iteration of the network to answer queries; data in a P2P database may well be distributed over several nodes and both networks – the XPeer network, as well as the Piazza network – assume scenarios in which a query results in a collection of data that is not only on a different peer in the network, but in

fact distributed (as in scattered) across the network [3], [8]; this is the incentive for providing the architecture for allowing complex data matching across multiple peers in a network, such as it is described for the Piazza PDMS [8].

Both of the Peer-to-Peer systems XPeer and Piazza initiate queries on a local peer, while for the XPeer project, it is mentioned that „a query is submitted by the user to a peer“ [3], [9].

The network traversal algorithm differs between XPeer and Piazza, since XPeer makes use of a hierarchical superpeer architecture; the superpeers are dynamically assigned (based on a resource availability assessment) and self-organized into a hierarchy, with leaf nodes possessing data; parent nodes are superpeers and possess children sets of leaf nodes or sets of further superpeers; each parent possesses a full overview of data available through its extended network of children, with a root node that possesses an entire view of the network [3].

A peer in the XPeer P2P database network – after successful parsing of an input query – proceeds to send the query to its immediate parent node in the XPeer architecture, which again sends to its own parent node, until the root node is reached; this occurs for each query and allows for a broader search of information in the network beyond the initial node which was queried, without traversing each single node in the system architecture [3]. In the opinion of the author, this method comes close to a centralized model, in which a single or a group of nodes perform administrative tasks essential to the system [2], [6]. However, it should be noted that the XPeer network has the ability to dyamically adapt to harsh challenges in database traversal which can disburden the superpeers from potentially high workloads; Sartiani et al. mention specifically the ability of superpeers to relocate children, delegate workload to other peers or even simply disconnect children if the previous options are unavailable [3].

The traversal algorithm of the Piazza PDMS is similar to the „flooded request model“ described in section II-D; a peer initiating a query, after reformulation is completed, forwards this query to all peers it is connected to [9]. Such a model obviously falls victim to the same drawback of the potential flooding of the entire network with each single query. A partial solution to this problem is to ignore peers who do not appear to be able to respond to a given query with appropriate results, based on the set of schema mappings that map the local data schemas to those of available peers; Tatarinov et al. also mention the key challenge of a „scalable index that returns, for any query, the set of relevant peers with as few false positives as possible“; a centralized indexing architecture for the Piazza PDMS was developed [9].

### *D. Security*

Security in P2P networks is a broad subject, ranging from communication with trusted parties to prevent contact with malicious users (on a low, technical level), to the ability to restrict sharing, for example for the protection of copyright, on a higher, social level [2], [6].

While security is not specifically mentioned in connection with the XPeer project, Tatarinov et al. mention security in the context of their Piazza PDMS as „an important consideration“: „Although the goal and emphasis of Piazza is data sharing, in practice, peers are almost never willing – or even legally able – to share their data in an uncontrolled way“ [9]. Global security policies are – due to the nature of the P2P network – not applicable. It is stated that the Piazza PDMS however implements a key exchange system, through which peers – referred to as „owners“ – may be able to control access to their data [9].

## IV. Incentives for Peer-to-Peer Databases

The incentives for employing peer-to-peer technology in the field of databases over the concept of more static distributed database systems is similar to the common reasons in favor of peer-to-peer technology outlined previously.

Sartiani et al. provide an overview of potential incentives for peer-to-peer databases.

> „The p2p paradigm was recently adopted in the database community to overcome the limitations of distributed database systems, namely the static topology and the heavy administration work, and to exploit the dissemination of data sources over the Internet“ [3].

Compared to the general advantages as detailed by Fischbach et al. in section II-E, [3] makes a point that P2P databases offer the same technological incentives of reduced administrative efforts due to the self-organization and easy extension.

Sartiani et al. also list the „dissemination of data sources over the Internet“ [3]; a peer-to-peer database might connect a variety of data sources that would otherwise be unavailable to each other. This interconnection and consolidation of sources is a main field of application of Peer-to-Peer databases, as projects such as SkyQuery may profit from the use of such databases in their effort to connect the various heterogenous sources of scientific data all over the world [9], [10]

There are clear incentives for the potential deployment of Peer-to-Peer databases; the following examples of applications that may make use of the concept support this claim.

## V. Applications

There is a wide area of applications for peer-to-peer databases. Sartiani et al. state that the XPeer project, for example, is a „general purpose XML p2p database system, so it can be used in any application field“ [3]. However, it must be acknowledged that as with other peer-to-peer concepts such as file sharing over a P2P network, the technology may not be useful in all application areas.

The fact that single autonomous units may join or leave the P2P network at any time [2], may render it unsafe to use in a context where the network user(s) rely on high availability of files (in the context of a filesharing network) or data (in the context of a database). Another concern may be the potential incompleteness of the query algorithms: employing techniques

to prevent traffic proliferation of fully decentralized network, like a TTL (time-to-live) as it is used in Gnutella, may lead to queries not reaching all possible nodes of interest [6].

Despite the challenges and drawbacks of peer-to-peer database systems, there is still a wide range of applications for the technology. Benefits of general peer-to-peer contepts include the simple extension of such networks, increased fault-tolerance or lower operational costs [2]. A very specific benefit of peer-to-peer databases may also be the easy extension of database networks by not only single nodes, but entire networks; as described in [11], several entire networks may be easily merged using the Piazza project, if a semantic schema mapping is created between nodes in each of the networks.

Sartiani et al. state that the XPeer system's „main application is the management of resource descriptions in a GRID-like environment: in particular, XPeer should form the basic infrastructure for extending (and, eventually, replacing) the LDAP-based resource discovery layer of existing GRID systems“ [3]. The Piazza Peer Data Management Project (PDMS) was designed with the goal to provide a system that enables „sharing heterogeneous data in a distributed and scalable way“ [9]. Halevy et al. states:

> The goal of the peer data management system is to address this need: we propose the use of a decentralized easily extensible data management architecture in which any user can contribute new data, schema information, or even mappings between other peers' schemas [11].

As an example for possible uses of the Piazza PDMS, Halevy et al. introduce a specific potential application as well, the implementation of the P2P database for emergency services across U.S. state borders: In the scenario, two groups – one being local police stations, hospitals and related institutions, the other one being emergency response organizations – make use of the P2P database system in their own domain; as described, in the event of an emergency (an earthquake is provided as an example), the emergency services would be able to immediately join the local P2P data network to effectively assist with ongoing operations, given the correct schema mappings are available [11]. Such an instant connection might significantly ease the ability of involved organizations to successfully access all required, key information without having to constantly maintain one single and large cross-departmental database.

[9] also mentions the SkyQuery project as an example of a project in which the technology of P2P databases may be of use. SkyQuery is an effort to join a large number of independent databases of autonomous and heterogeneous nature [10]. Malik et al. mention other scientific fields demanding similar consolidation, specifically pharmaceutical, medical or geographical fields [10]. At last, the Semantic Web is mentioned as a project that may be supported by peer-to-peer databases [9].

## VI. Conclusion

The paper has outlined the basic technology and common concepts of modern P2P networks. It showed how these concepts relate in detail to P2P databases. While some applications of peer-to-peer networks are already in use – such as the widely discussed file-sharing platforms [1] – pure peer-to-peer databases have to the best knowledge of the author not yet transcended into mainstream adoption.

The broad theoretical concepts of peer-to-peer databases have been discussed in detail. The specific challenges of P2P databases include the autonomy of single nodes in the network, the mapping of schemas of local nodes to foreign nodes, the querying of data in the networked database without a central authority and security aspects. The key technological problem as perceived by the author, is the apparent lack of a complete and efficient query algorithm in combination with a pure P2P network.

Benefits of P2P networks include especially reduced administrative efforts due to the self-organization and the easy extension of the network by the addition of new nodes [3] or even entire new networks by means of defining new schema mappings [11]. This eliminates some of the main drawbacks of classic alternatives of networked databases, such as their „static topology“ and „heavy administration work“ [3].

The paper described two P2P database projects, the Piazza PDMS and the XPeer Project. Both of these systems provide a networked database, with stored data in the form of XML forests and mention similar drawbacks such as potentially incomplete queries, difficulties with sudden changes in network topology and potential security issues [3], [9].

## References


[1] D. Schoder and K. Fischbach, "Peer-to-peer prospects," *Commun. ACM*, vol. 46, no. 2, pp. 27–29, 2003.

[2] K. Fischbach, C. Schmitt, and D. Schoder, "Core concepts in peer-to-peer networking," in *Peer to Peer Computing: The Evolution of a Disruptive Technology*. Idea Group Inc., 2005, ch. 1. [Online]. Available: http://www.idea-group.com/downloads/excerpts/Subramanian01.pdf

[3] C. Sartiani, P. Manghi, G. Ghelli, and G. Conforti, "Xpeer: A self-organizing xml p2p database system," in *In Proceedings of the First EDBT Workshop on P2P and Databases (P2P&DB)*, 2004, pp. 456–465.

[4] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 1–26, 2008.

[5] S. Gribble, A. Halevy, Z. Ives, M. Rodrig, and D. Suciu, "What can databases do for peer-to-peer?" in *In WebDB*, 2001.

[6] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, "Peer-to-peer computing," 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.131.9873

[7] A. Kementsietsidis, M. Arenas, and R. J. Miller, "Mapping data in peer-to-peer systems: Semantics and algorithmic issues," 2003.

[8] A. Halevy, Z. Ives, P. Mork, and I. Tatarinov, "Piazza: Data management infrastructure for semantic web applications," 2003, pp. 556–567.

[9] I. Tatarinov, Z. Ives, J. Madhavan, A. Halevy, D. Suciu, N. Dalvi, X. L. Dong, Y. Kadiyska, G. Miklau, and P. Mork, "The piazza peer data management project," *SIGMOD Rec.*, vol. 32, no. 3, pp. 47–52, 2003.

[10] T. Malik, A. S. Szalay, T. Budavari, and A. R. Thakar, "Skyquery: A web service approach to federate databases," in *In Proc. CIDR*, 2003.

[11] A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov, "Schema mediation in peer data management systems," in *In ICDE*, 2003, pp. 505–516.

# IT-Sicherheit - Faktor Mensch und psychologische Aspekte

Simon Stauber
Betreuerin: Corinna Schmitt
Seminar Innovative Internettechnologien und Mobilkommunikation SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: stauber@in.tum.de

***Kurzfassung*— IT-Sicherheit basiert nicht nur auf technischen Aspekten, sondern ist in hohem Maße dem Verhalten der Menschen bezüglich Datensicherheit unterworfen. Das Handeln einzelner Individuen wird durch psychologische Faktoren beschrieben und basiert deswegen für gewöhnlich auf irrationalen Entscheidungen. Diese Gegebenheit wissen Angreifer auf IT-Systeme für ihre Zwecke zu verwenden, indem sie die psychologischen Verhaltensmuster der Anwender in bestimmten Situationen ausnutzen. Auf diese Weise scheitert IT-Sicherheit am Faktor Mensch, der als das schwächste Glied in der Sicherheitskette gesehen wird und somit durch Fehlverhalten einen Großteil der technischen Sicherheitsmechanismen unterwandert. Folglich soll der Mensch auf Faktoren der IT-Sicherheit sensibilisiert werden, so dass IT-Systeme umsichtig genutzt und vertrauliche Daten geschützt werden.**

***Schlüsselworte*— IT-Sicherheit, menschliches Verhalten, psychologische Aspekte, Awareness, Sicherheitsbewusstsein, Privatsphäre, Entscheidungsfindung, Erwartungstheorie, Besitztumseffekt, Datenschutz, Social Engineering, Phishing, Social Network Sites**

## I. Einleitung

Die Kommunikation und der Austausch von Daten über technische Systeme hat in den letzten Jahren stark an Bedeutung gewonnen und ist einem stetigen Wachstum unterworfen. Die rasche Entwicklung der Telekommunikationssysteme wie Mobiltelefonnetze, aber auch die Verwendung des Internets dienen als Basis für das Versenden und Bereitstellen von Informationen. Häufig werden über diese Kanäle Vorgänge abgewickelt, welche die Übermittlung von sensiblen und schützenswerten Daten nötig machen, die unter keinen Umständen an dritte Personen herausgegeben werden dürfen. Beispielhaft sind hier das Online-Banking oder auch der Einkauf mittels Kreditkarte bei einem Online Versandhaus zu nennen.
Hierfür reicht es nicht aus, dass die Technik gewissen Sicherheitsstandards unterliegt. Auch der Mensch als Benutzer der Kommunikationsmittel muss durch sein Verhalten dazu beitragen, dass vertrauliche Informationen geschützt werden. Diese Faktoren werden unter dem Begriff IT-Sicherheit zusammengefasst, welcher näher erläutert werden soll. Hierbei wird sowohl die aktuelle Sicherheitsphilosophie der Hard- und Softwarehersteller beleuchtet, als auch das allgemeine Sicherheitsbewusstsein der Benutzer beschrieben und auf welche Weise diese bezüglich Datenschutz sensibilisiert werden können. Dazu wird auf den Terminus der Privatsphäre eingegangen, wobei aufgezeigt werden soll, welchen psychologischen Einflüssen Menschen unterliegen, damit die Bereitschaft geweckt wird, die Privatsphäre einzugrenzen oder sogar ganz aufzugeben.
Diese Einflüsse können zum Datendiebstahl durch Anwendung von Social Engineering Techniken verwendet werden, wobei durch Ausnutzung menschlicher Verhaltensweisen schützenswerte Informationen erschlichen werden können. Neben dieser Vorgehensweise, welche von Seiten des Angreifers meist Gewandtheit im Umgang und Kommunikation mit Mitmenschen erfordert, soll anschließend auf Social Network Portale eingegangen werden. Sie sollen als Fallbeispiel dienen und veranschaulichen, wie unkompliziert und schnell durch das Missachten von IT-Sicherheit persönliche Daten an völlig fremde Personen gelangen können und welche Folgen sich aus diesem Umstand ableiten.

## II. Säulen der IT-Sicherheit

Das Konzept der IT-Sicherheit basiert auf drei Säulen [1]:
Die erste umfasst die Aspekte der *Technik* und sieht alle Maßnahmen vor, welche durch Einsatz von Hard- und Softwarekomponenten zur Absicherung eines IT-Systems führen. Hierunter fällt das Firewallkonzept, welches das interne Netzwerk und dessen Komponenten vor unerlaubten Zugriffen schützen soll. Ergänzt durch die Wahl eines passenden Betriebskonzepts soll ein effektiver Abwehrmechanismus gegen Schadsoftware wie Viren oder Trojaner, aber auch Spamnachrichten geschaffen werden. Durch Monitoring soll es möglich gemacht werden, das IT-System in einer Form zu überwachen, dass ein bösartiges Eindringen in das Netzwerk sofort erkannt wird. Der letzte Punkt sieht ein Notfallsystem vor, das zum einen in der Lage ist eine Datenwiederherstellung zu initiieren, aber auch ein Fallback System starten kann.
Die zweite Säule bezeichnet die *Organisation*. Dabei sollen die Zuständigkeitsbereiche einzelner Personen für die jeweiligen Aufgabengebiete festgelegt werden, damit ein optimales Sicherheitskonzept des IT-Systems erarbeitet und ausgeführt werden kann. Dies sieht nicht nur technische Aspekte vor, sondern soll auch passende Prozeduren bereithalten, welche in Fehlerfällen einzuhalten sind, beispielsweise in Form eines

Notfallplans. Dies schließt auch die Erstellung von verschiedenen Richtlinien mit ein, die etwa Backupintervalle oder eine Zugriffsverwaltung darlegen sollen.
Diese erstellten Policies sind letztendlich vom *Menschen*, der dritten Säule der IT-Sicherheit, zu beachten. Es gilt den Benutzern des IT-Systems ein Verständnis für die technischen Grundlagen zu geben, damit darauf basierend eine Akzeptanz der Richtlinien eintritt und die Anwendung von sicherheitsrelevanten Vorgängen getätigt wird. Dies kann beispielsweise durch Schulungen geschehen, welche die sogenannte Awareness der Menschen erhöhen soll.
Aus dem Blickwinkel der Psychologie ist der Begriff der IT-Sicherheit auf ähnliche Weise untergliedert, indem drei verschiedene Gefährdungsepochen beschrieben werden [2]:
*Physikalische* Gefährdungen thematisieren Sicherheitslücken und Fehlfunktionen, die auf Hardwarebasis einzuordnen sind.
*Syntaktische* Gefährdung entsteht unter der Verwendung bzw. durch die Erstellung von fehlerhaften Softwarekomponenten, deren Anfälligkeit ausgenutzt werden kann, um letztendlich das Sicherheitskonzept des IT-Systems zu umgehen.

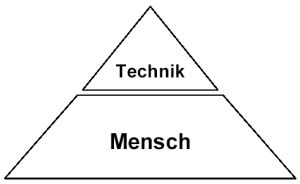



Abbildung 1. Symbolhafte Darstellung des Sicherheitsaspekts bei der Abhängigkeitsbeziehung Mensch und Technik

Als *semantische* Gefährdung wird der Mensch und dessen Verhalten bezeichnet. Er kann durch sein fehlerhaftes Handeln die auf technischem Weg hergestellte Sicherheit eines Systems zerstören, beispielsweise durch die unerlaubte Herausgabe von Passwörtern. Hierdurch wird der Mensch als grundlegender Faktor der IT-Sicherheit eingestuft, wie Abbildung 1 zeigt, der durch sein inkorrektes Handeln alle anderen Sicherheitsmaßnahmen außer Kraft setzen kann. Diesen Schluss hat Degenhardt [2] aufgezeigt, wobei diese Gefährdung durch die Sensibilisierung von Benutzern (Awareness) verringert werden kann.

## III. Awareness

Damit IT-Sicherheit erhöht werden kann, ist es notwendig die Gefahren, welche durch das fehlerhafte Verhalten von Menschen ausgehen, zu minimieren. Hierzu soll die Aufmerksamkeit der Benutzer von IT-Geräten im Bezug auf sicherheitstechnische Aspekte geschult werden. Dieses Prinzip der Sensibilisierung wird als *Awareness* bezeichnet.
Esslinger [1] stellt für eine erfolgreiche praktische Anwendung von Awareness vier Aspekte in den Vordergrund:

- Menschen müssen eine gewisse *Motivation* aufbringen, damit sie persönliche oder firmeninterne Informationen als schützenswert betrachten. Dazu muss sowohl die Wertigkeit der erzeugten Daten verinnerlicht, als auch die Folgen bei einem Diebstahl verdeutlicht werden. Der Benutzer muss Aufmerksamkeit gegenüber den zahlreichen Möglichkeiten von Datenklau entwickeln.

- *Richtlinien* sollen den Benutzer durch die Vorgabe von konkreten Vorgehensweisen unterstützen, das Verhalten anzupassen, damit eine möglichst umfangreiche IT-Sicherheit gewährleistet werden kann. Beispielsweise soll somit die Herausgabe oder ungesicherte Ablage von Passwörtern vermieden werden.

- Über *Feedback und Training* werden Menschen über aktuelle Gefahren in der IT informiert. Dazu ist es unabdingbar, dass, etwa bei Firmen, in regelmäßigen Abständen, passende Informationen bereitgestellt werden. Darüber hinaus sollte der Benutzer im Idealfall selbsttätig durch aufmerksames Verfolgen der Medien auf einem aktuellen Stand bezüglich IT-Sicherheit gehalten werden.

- *Benutzbarkeit* beschreibt die unkomplizierte Handhabung von sicherheitsrelevanten Vorgängen wie Datenverschlüsselung. Menschen werden Maßnahmen nur akzeptieren und anwenden, wenn sie dadurch nicht wesentlich in ihrer Arbeit beeinträchtigt werden. Auf diese Problemstellung geht auch Degenhardt [3], [4] ein, wie in Abschnitt IV-B beschrieben wird.

Abschließend kann festgehalten werden, dass menschliches Verhalten einen bedeutenden Anteil bei jeder Sicherheitsmaßnahme aufweist.

## IV. Sicherheitsbewusstsein

IT-Sicherheit fordert sowohl Sicherheitsbewusstsein von Softwareherstellern, als auch von den Menschen, die deren Produkte verwenden. Da beide Parteien auf unterschiedliche Art und Weise handeln müssen, damit eine geschützte IT-Infrastruktur erzeugt wird, ist es sinnvoll, die Analyse ihres Verhaltens zu trennen.

### A. Hersteller

Die Hauptaufgabe der Hersteller von IT-Komponenten wie Softwareprodukten besteht darin, diese frei von Mängeln und sicherheitskritischen Fehlfunktionen auszuliefern. Dies trifft in der Realität oftmals nicht zu, wie Degenhardt in [2], [4] thematisiert.
Software wird gewöhnlich noch mit Mängeln bestückt an die Kunden weitergegeben. Dies ist teils bedingt durch den Konkurrenzdruck anderer Hersteller, welcher einen schnellen Release der Produkte erfordert. Zusätzliches gewissenhaftes Prüfen und Verbessern der Software treibt die Kosten oftmals massiv in die Höhe. Diesen Preis ist der Benutzer meist nicht gewillt zu bezahlen. So ist es für den Endverbraucher beinahe zur Gewohnheit geworden auf zeitversetzte Nachbesserungen zu warten, um damit die bedenklichsten Sicherheitslücken zu

schließen.
Außerdem wird speziell Software als „nicht-substituierbares Gut“ [2] bezeichnet, das in seiner Lebensdauer prinzipiell keinen Einschränkungen unterworfen ist. Aus diesem Grund ist es dem Hersteller nur möglich durch neue Programmversionen, welche Verbesserungen und neue Funktionalitäten beinhalten, weiteren Gewinn zu erzielen. Dieses Prinzip wird als „Anreiz-Perversion“ [2] bezeichnet.
Sichere und verifizierte Software wird allerdings nicht ausgeschlossen. Auf Grund des bereits angesprochenen Zeit- und Kostenfaktors wird Qualitätssoftware hauptsächlich in von Degenhardt angesprochenen „Ökosysteme“ [3] eingegrenzt. Darin wirken rechtliche, finanzielle oder ethische Faktoren, die den Hersteller nötigen, auf sicherheitstechnische Aspekte Wert zu legen. Als Beispiel ist die Steuerung von Atomkraftwerken, öffentlichem Nahverkehr oder auch Passagierflugzeugen zu nennen. Hier können fehlerhafte Komponenten zu katastrophalen Auswirkungen führen, weshalb etwa entsprechende zuständige Software oftmals nur die nötigste Funktionalität aufweist, diese jedoch als verlässlich betrachtet werden kann.

### B. Benutzer

Der Benutzer ist gewöhnlich auf die Nutzung der vom Hersteller erzeugten Komponenten beschränkt. Menschen wollen möglichst schnelle Datenverarbeitung vollziehen und sind deswegen dazu geneigt, vermeintlich überflüssige, aber zeitaufwendige Prozeduren zu meiden, auf Kosten von sicherheitstechnischen Einbußen. Aus diesem Grund wird beispielsweise auf Verschlüsselung verzichtet, Benutzer verschieben wichtige Updates des Betriebssystems oder ignorieren Warnmeldungen [4].
Dieses Verhalten wird aus verschiedenen Gründen hervorgerufen. Gerade im Umfeld von Firmen stehen Mitarbeiter oftmals unter Zeitdruck und agieren, bedingt durch Stress, nicht auf rationale Weise, sondern arbeiten zielstrebig nur den brisantesten Aufgabenpunkt ab. Zur Sicherung des IT-Systems und für aufmerksames Verhalten kann deshalb wenig Zeit veranschlagt werden.
Neben Zeitmangel ist fehlende Kompetenz für IT-Systeme ein weiterer relevanter Aspekt. Folglich ist Benutzern die Tragweite ihres Handelns durch Mangel an Hintergrundwissen nicht bewusst und Sicherheitslücken werden geöffnet ohne Einfluss von böswilliger Absicht. Degenhardt fasst diese Problematik unter dem Stichwort „Medienkompetenz“ [3] zusammen. Werden ungeschulte Benutzer mit derartigen Situationen konfrontiert, sind sie nicht in der Lage rational zu denken, sondern treffen eine Gefühlsentscheidung.
Letztendlich laufen Menschen Gefahr, Bedrohungen durch unsichere IT-Systeme zu verharmlosen, da die entstehenden Folgen im digitalen Umfeld nicht direkt greifbar sind [3]. Deswegen stellt das Sicherheitsbewusstsein für den Benutzer nur einen sekundären Aspekt dar, dessen Bedeutung durch Awareness, die in Abschnitt III thematisiert ist, gesteigert werden muss.

## V. Privatsphäre

Der Schutz und sensible Umgang mit der Privatsphäre eines Menschen hilft die IT-Sicherheit zu festigen, da durch das Bewahren persönlicher Daten der Zugang für dritte, unbefugte Personen zu IT-Systemen erschwert wird. Dies kann beispielhaft an einem Email Benutzerkonto verdeutlicht werden. Der Benutzer sollte an dieser Stelle seine Privatsphäre nicht nur durch die Geheimhaltung seines Passworts bewahren, sondern auch durch den sensiblen Umgang mit privaten Daten, welche oftmals als Zugangsschutz dienen, falls das eigentliche Passwort vergessen wurde. Dieser Gesichtspunkt spielt auch beim Social Engineering (Abschnitt VI) eine erhebliche Rolle.
Obwohl nach Prechelt [5] die Definition von Privatsphäre nicht eindeutig festlegbar ist, beschreibt er sie dennoch als den Bereich, in dem eine Person selbst bestimmt (oder bestimmen könnte), wem sie wann und warum welche Informationen über sich selbst zugänglich macht.
Folglich zeigt sich, dass Menschen ihre Privatsphäre sorgsam behandeln sollten und genau abwägen, welche privaten Informationen sie Mitmenschen zur Verfügung stellen.

### A. Entscheidungsfindung

Letztendlich müssen Menschen den Prozess der Entscheidungsfindung durchlaufen, um einen Entschluss über die Herausgabe ihrer Daten zu treffen. Dabei werden sie verschiedenen Einflussfaktoren unterworfen, die im folgenden erläutert werden.
Wie Weichert [6] thematisiert, ist es nicht möglich in allen Situationen seine Privatsphäre zu bewahren, da man durch das Verhalten im täglichen Leben Spuren hinterlässt. Dies äußert sich beispielsweise bei der Nutzung des Internets. Hier können Provider Rückschlüsse auf das Surfverhalten des Kunden ziehen. Allerdings lässt sich dieser Aspekt nicht nur auf die „digitale Welt“ eingrenzen. Auch der Aufenthalt an öffentlichen, kameraüberwachten Plätzen kann unbekannten Personen Informationen wie Ort, Zeit, ausgeführte Handlung und Körpermerkmale liefern.
Neben diesen kaum vermeidbaren Faktoren ist der Mensch psychologischen Aspekten unterworfen, die seine Entscheidung beeinflussen, welche zu irrationalen Verhalten beim Schutz von privaten Daten führen kann. Acquisti [7] zeigt Aspekte auf, welche sich in diesen Bereich einordnen lassen:
Menschen glauben an die *Kontrollmöglichkeit* ihrer persönlichen Daten, auch nachdem sie bereits anderen Personen zur Verfügung gestanden haben. Werden beispielsweise aus Online-Portalen von Versandhäusern oder Social Network Sites persönliche Informationen wie Adresse, Hobbies oder Kreditkartennummer zurückgezogen, wird das Rückholen der Daten suggeriert, obwohl diese bereits auf anderen Kanälen verbreitet worden sein könnten.
Bei einer häufigen *Konfrontation* mit bestimmten sicherheitskritischen Vorgängen werden diese als „gewöhnlich“ eingestuft und verlieren für den Menschen an Wichtigkeit. So stellt es für den Benutzer oftmals keine Außergewöhnlichkeit dar, seine Kreditkartennummer anzugeben, da zahlreiche

geschäftliche Vorgänge im Internet damit abgewickelt werden.
Darüber hinaus ist man dazu geneigt, den Sicherheits- und Datenschutzaspekt eher außer Acht zu lassen, wenn man sich dadurch einen großen *Vorteil* verspricht. Zahlreiche Leute geben durch Payback-Karten ihr Kaufverhalten Preis, da sie dadurch einen Rabattvorteil genießen.
Acquisti [7] zeigt auf, dass bei der Entscheidungsfindung zwei wesentliche Elemente eine Rolle spielen: Zunächst soll festgestellt werden, welche datenschutzrelevanten Ereignisse auftreten. Anschließend kann geprüft werden, wie weit die daraus resultierenden Konsequenzen führen. Weichert [6] leitet daraus ab, dass Menschen für die Weitergabe von privaten Daten eine sinnvolle Abschätzung zwischen Risiko und Gewinn vollziehen müssen. Acquisti [7] unterscheidet an dieser Stelle zwei Begrifflichkeiten: *Risiko* und *Unsicherheit*. Risiko beschreibt die Existenz einer Wahrscheinlichkeit zu einem auftretenden Ereignis, während Unsicherheit die Unmöglichkeit beschreibt, die Wahrscheinlichkeit vorherzubestimmen. Diese Unsicherheiten entstehen neben den bereits erwähnten psychologischen Faktoren noch durch weitere Einflüsse:

- Menschen besitzen oftmals *begrenztes Wissen* über die Möglichkeiten des Datenschutzes und verstehen die Tragweite bei der Herausgabe von Informationen nicht. Zusätzlich ist es meist nicht möglich nachzuvollziehen, welche Verwendung die preisgegebenen Informationen finden und somit ist die resultierende Konsequenz schwer zu erkennen.

- Das Eintreffen von *Ereignissen in der Zukunft* ist unvorhersehbar. Beispielsweise durch die Schaffung neuer Technologien im Bereich der Suchmaschinen können ehemals schwer auffindbare Informationen heute mühelos entdeckt werden.

Letztendlich muss versucht werden, das entstehende Risiko abzuschätzen, welches durch das Modell der *Ankerung* und *Anpassung* beschrieben ist. Es besagt, dass sich Individuen durch eine initiale Abschätzung ein ungefähres Bild des Risikos verschaffen (Ankerung). Dieses wird dann durch die Abwägung anderer Alternativen und Erfahrungen angepasst (Anpassung).

### *B. Theorien zu irrationalem Datenschutzverhalten*

Anhand von verschiedenen Theorien und Studien kann aufgezeigt werden, dass Menschen irrationale Entscheidungen treffen. Obwohl diese Theorien teilweise der Wirtschaftsökonomie entstammen, ist es möglich diese auf den Schutz privater Daten bzw. der Privatsphäre im Bezug auf IT-Sicherheit abzuleiten.
In vorherigem Abschnitt wurde erläutert, dass es unmöglich scheint, das reale existierende Risiko korrekt zu berechnen, um daraus eine richtige Verhaltensweise resultieren zu lassen, da unbekannte Faktoren und Konsequenzen auftreten können. Simon [6], [8], [9] zeigt durch den Begriff der *eingeschränkten Rationalität*, dass sogar nicht zwingend das korrekte Verhalten ausgelöst wird, wenn der handelnden Person alle Informationen bezüglich der resultierenden Konsequenzen vorliegen würden. Auf Grund von Zeitmangel und seiner limitierten Aufnahmefähigkeit hat der Mensch nur einen begrenzten Zugang zu den verfügbaren Ressourcen. So können nicht alle Möglichkeiten betrachtet werden, da die Wahlalternativen durch die Wahrnehmung gefiltert werden. Entsprechend wird der Mensch als „Satisficer“ bezeichnet, da er nur solange nach einer passenden Alternative sucht, bis diese ein zufriedenstellendes Ergebnis darstellt.
Die *Erwartungstheorie* zeigt das Verhalten von Personen bei der Aussicht auf einen möglichen *Gewinn* oder *Verlust*. Sie sagt aus, dass Menschen bei der Aussicht auf einen geringen, sicheren Gewinn weniger Risiko eingehen und dadurch lieber auf die Möglichkeit eines eventuellen hohen Gewinns verzichten. Steht jedoch ein Verlust zur Debatte, neigen Individuen dazu, mehr Risiko auf sich zu nehmen, um den drohenden Verlust abzuwenden, auch wenn die Gefahr besteht, dass dieser letztendlich noch größer ausfällt.
Weichert [6] beschreibt diese Theorie anhand eines Beispiels: Darf eine Versuchsperson zwischen einem sicheren Gewinn von 500 Euro oder einer 50% Chance auf einen Gewinn von 1000 Euro entscheiden, so wird sie meist die erste Alternative wählen. Wird die Person mit selbiger Situation auf einen Verlust konfrontiert, wird sie sich für die risikoreiche 50% Variante entscheiden.
Schneier [10] stellt klar, dass ein rein ökonomisches, mathematisches Modell hier keinen Unterschied zwischen Gewinn und Verlust aufzeigen könnte. Eine Erklärung für das irrationale Verhalten der Versuchspersonen wird an der Evolution festgemacht, da die vielversprechendere Überlebensstrategie an kleine und sichere „Gewinne“ gekoppelt ist. Schneier [10] erklärt durch die Erwartungstheorie das Verhalten der Benutzer mit dem Umgang von sicherheitsbewahrenden Produkten:
So finden beispielsweise IT-Produkte zur Absicherung eines Netzwerks geringen Anklang beim Benutzer, da die entstehenden Kosten einen kleinen sicheren Verlust darstellen, gegenüber einem nur unter Umständen auftretenden hohen Verlust, der durch Datendiebstahl und Bruch der Privatsphäre entstünde. Da es sich dabei um eine Verlustsituation handelt, bevorzugt der Mensch an dieser Stelle die risikoreiche Variante.
Darüber hinaus sieht Acquisti [7] noch zusätzliche Einflüsse, die sich in Form von Desinteresse am Schutz der eigenen Privatsphäre und Sorglosigkeit gegenüber der Zukunft äußern.
Ein Effekt, der im Bezug zur Erwartungstheorie steht, ist der *Besitztumseffekt* [7]. Dieser besagt, dass Menschen ein Gut als wertvoller einschätzen, wenn sie im Besitz von diesem sind. Wollten sie selbiges Gut erwerben, wird der Wert des Gutes als geringer empfunden und ein niedriger Kaufpreis angesetzt. Kahneman [11] beschreibt diesen Effekt anhand eines Experiments: Personen wurden in zwei Gruppen eingeteilt. Die erste erhielt eine Kaffeetasse und wurde beauftragt für diese einen Verkaufspreis zwischen 0,25 US$ und 9,25 US$ festzulegen.

Da sie im Besitz der Tasse waren, wollten sie für diese ein hohen Preis von durchschnittlich 7,12 US$ erzielen. Die zweite Gruppe sollte die Kaffeetasse erwerben und entsprechend einen für sie akzeptablen Kaufpreis festlegen. Da sie nicht im Besitz des Gutes waren, fiel der durchschnittliche Preis auf geringe 2,87 US$ aus.
Zeichick [12] stellt fest, dass Menschen kaum die Neigung zeigen, in ihrem Besitz befindliche Güter gegen ähnlich Dinge auszutauschen, obwohl diese einen etwas höheren Marktwert erzielen. Individuen wollen dadurch den Faktor des neuen und Unbekannten minimieren, da nicht vorhergesagt werden kann, ob das neue Gut ebenfalls die gleichen, zuverlässigen Eigenschaften aufweist wie das alte. Darüber hinaus haben Menschen meist in ihre Entscheidung für den Kauf eines Gutes investiert, so dass sie dieses nicht einfach ziellos gegen ein anderes umtauschen möchten.
Dieses Verhalten kann auch auf die Verwendung von IT-Produkten projiziert werden. Haben sich Benutzer an spezielle Software oder technische Geräte gewöhnt, so ziehen sie es vor, Veränderungen an diesen zu vermeiden, um den Status Quo zu erhalten. Aus diesem Grund werden sicherheitsnotwendige Updates oder der Austausch von alten Programmversionen bewusst zurückgehalten oder vermieden. Der Mensch hat sowohl eine emotionale Bindung zu den erworbenen Produkten und scheut dadurch den Wechsel zu neuen ungewohnten Umgebungen, als auch eine finanzielle Bindung, die den Kauf neuer Produkte mit ähnlicher Funktionalität hemmt.
Abschließend bleibt festzustellen, dass Individuen nach der Beschreibung verschiedener Verhaltensmodelle agieren und dadurch irrational handeln. Hieraus entsteht die Gefährdung privater Daten, der Privatsphäre und der IT-Sicherheit durch den Menschen.

## VI. Social Engineering

Im vorherigen Abschnitt wurde dargestellt, dass der Mensch psychologischen Aspekten unterworfen ist und sein Verhalten daraus ableitet. Die Schwachstelle des irrationalen Handelns von Individuen kann von Angreifern ausgenutzt werden, um unberechtigterweise an vertrauliche Informationen zu gelangen und dadurch die IT-Sicherheit zu untergraben.
Hierfür wird die Technik des Social Engineerings angewendet. European Network and Information Security Agency (ENISA) [13] definiert dies als Technik, menschliche Schwächen und Manipulationsfähigkeit auszunutzen, um Sicherheitsmechanismen zu brechen. Ziel ist es an sensible Daten und Informationen einer dritten Partei zu gelangen, um diese in betrügerischer Absicht zu verwenden.
Die Motivation von Social Engineers ist vielfältig. Oftmals soll ein finanzieller Vorteil erlangt werden, beispielsweise durch das Erschleichen von fremden Bankdaten und Kreditkarteninformationen. Grafe [14] nennt weiter Punkte, unter die Industriespionage, Identitätsdiebstahl, Spass- und Machtfaktor, sowie soziale Gründe fallen. Letztere beziehen sich kaum auf den finanziellen Aspekt, sondern vielmehr auf negative gesellschaftliche Einflüsse, wie Stalking oder Mobbing, welche im Abschnitt VII behandelt werden.

### A. Psychologische Aspekte

Social Engineers verwenden die Eigenschaften der menschlichen Psychologie gegen ihre Opfer, indem sie mit ihnen in Kontakt treten. Meist geschieht dies durch fingierte Anrufe, wobei der Angreifer eine falsche Identität vorgibt. Dabei nutzen Angreifer, laut ENISA [13], folgende Verhaltensweisen des Opfers aus:
*Berechtigung* beschreibt den Vorgang, in welchem sich der Angreifer als autorisierte Person ausgibt. Dies gelingt meist durch den Einsatz von bereits erlangten, scheinbar wertlosen Informationen, wie einer Personalnummer, Namen von Arbeitskollegen und Vorgesetzten oder das reine Wissen über den Ablauf gewisser firmeninterner Vorgänge. Wird das Opfer damit konfrontiert, wird der Angreifer leicht als berechtigte Person angesehen, da er bereits über vermeintlich interne Informationen verfügt. Um die Wirkung noch weiter zu steigern, gibt sich der Social Engineer als ranghöhere Person aus, so dass das Opfer Hemmungen bekommt, sein Anliegen abzulehnen. Mitnick [15] betont, dass das natürliche Bedürfnis behilflich zu sein, sich vervielfacht, wenn die Person, der man hilft, wichtigen Einfluss besitzt. Auf diese Weise wird eine eigentlich bedeutungslose Kommunikation bedeutungsvoll.
*Beständigkeit* behandelt die Eigenschaft, dass eine Zielperson weniger Verdacht auf Betrug hat, wenn ein wiederkehrendes Verhalten vorliegt. Dies kann erreicht werden, indem ein Angriff in mehrere Teilangriffe zerlegt wird, welche aufeinander aufbauen. Beim Opfer stellt sich hier der Effekt der Gewöhnung ein und Verdachtsmomente werden minimiert. Mitnick [15] beschreibt in einem Fallbeispiel die Kommunikation eines Social Engineers mit einem Mitarbeiter einer Videothek. Der Angreifer meldet sich hier regelmäßig via Telefon und gibt vor in einer anderen Filiale der Videothek beschäftigt zu sein. Sie führen lockere Gespräche, wobei dem Opfer kontinuierlich Informationen über den Arbeitsablauf entlockt werden, so dass in einem finalen Angriff die Kreditkartennummer eines Kunden herausgefunden wird.
Die zahlreichen Dialoge mit der Zielperson dienen jedoch nicht nur der Erlangung von Informationen. Gerade in der frühen Phase eines Angriffs soll durch bedeutungslose Kommunikation ein Vertrauensverhältnis aufgebaut werden. Dieser Vorgang wird unter dem Terminus *Zuneigung* beschrieben. Dieser besagt, dass Personen eher mit Menschen kooperieren, die sie mögen und sich auf ähnliche Weise verhalten. Beispielsweise gibt der Social Engineer bei den Gesprächen vor, die selbe Meinung mit der Zielperson zu teilen oder ähnliche Probleme und Anliegen zu haben. Das Opfer fühlt sich dadurch „verstanden“ und ist in der Lage ein Sympathieverhältnis aufzubauen.
Bei der *Wechselwirkung* wird der Zielperson etwas gegeben, so dass ein Gefühl der Verpflichtung zur Erwiderung gegenüber dem Social Engineer entsteht. Damit kann das Reverse Social Engineering in Bezug gebracht werden, wie es Grafe [14] thematisiert: Dazu bietet der Angreifer dem Opfer seine Hilfe zu einem vorgetäuschten Problem an und

kann dadurch in die Privatsphäre der Zielperson eindringen. Zusätzlich zeigt sich das Opfer durch die Unterstützung aufgeschlossener, eine Gegenleistung zu erbringen, welche der Social Engineer zu seinen Gunsten zu nutzen weiß.
*Knappheit* bezeichnet den Effekt, wenn vermeintlich Erwünschtes als begrenzt und kurzzeitig erhältlich ausgegeben wird. Liegt dies vor, wird beim Opfer ein überstürztes Handeln ausgelöst. Die geringe Bedenkzeit führt zu unüberlegtem und irrationalem Verhalten. Ein ähnlicher Effekt tritt ebenfalls ein, wenn das Opfer, beispielsweise in einem Telefongespräch, unter Zeitdruck gesetzt wird, indem der Angreifer vorgibt, gewisse Informationen umgehend für seinen Vorgesetzten einholen zu müssen.
Menschen benötigen oftmals eine *Bestätigung*, um festzustellen, ob ihre Entscheidungen korrekt sind. Deswegen wird eine Zielperson durch das Verhalten anderer Personen beeinflusst und passt sich entsprechend an. Eine hieraus entstandene „Gruppendynamik“ verschafft das Gefühl richtig gehandelt zu haben.
Es ist zu erkennen, dass Social Engineers eine Vielzahl an psychologischen Aspekten des menschlichen Verhaltens ausnutzen, um die IT-Sicherheit und den Schutz der Privatsphäre zu unterwandern, so dass sie an die gewünschten Informationen gelangen. Dazu treten sie meist in den Dialog mit dem Opfer.

### *B. Phishing*

Neben dem direkten Dialog mit dem Opfer besteht die Möglichkeit durch manipulierte elektronische Nachrichten mit diesen in Kontakt zu treten, um an vertrauliche Daten zu gelangen. Dies geschieht gewöhnlich mit sogenannten Phishing Mails. Sie beruhen auf der Technik, dass dem Opfer ein vertrauenswürdiger Absender vorgetäuscht wird, der mit der Bitte auf Rücksendung von persönlichen Informationen auftritt. Diese Daten sollen meist über ein gefälschtes Webformular, beispielsweise einer Bank, aber auch mittels Telefon übersendet werden. Auf Grund einer akkuraten Gestaltung der Internetseite und eines gehobenen Sprachstils, sind naive Opfer dazu geneigt, kein Misstrauen zu schöpfen.
Phishing wird im erweiterten Sinne zu Social Engineering gezählt, da es ebenfalls menschliche Verhaltensweisen ausnutzt, um an Informationen zu gelangen. Jedoch handelt es sich hierbei um einen hauptsächlich technischen Weg, der durch Zuhilfenahme von Schadsoftware, beispielsweise Trojanern, von einer psychologischen Wechselwirkung zwischen Opfer und Angreifer abrückt.

### *C. Social Engineering Zyklus*

Social Engineering stellt für gewöhnlich keinen alleinstehenden Vorgang dar, sondern ist durch einen kompletten Zyklus beschrieben, wie ihn Mitnick [15] aufzeigt:

- Zu Beginn steht die *Recherche*. Dabei handelt es sich um eine Suche nach initialen Informationen über die Zielperson, welche aus öffentlichen Quellen stammen. Beispielsweise können Zeitungsausschnitte, Webseiten, Berichte oder Telefonbücher zu Rate gezogen werden. Eine weitere Möglichkeit an Daten zu gelangen, besteht im Dumpster Diving, welches den Vorgang des Durchsuchens von Abfall und Mülleimern des Opfers bezeichnet.

- Als zweiter Schritt soll eine *Beziehung und Vertrauen* zwischen Angreifer und Opfer aufgebaut werden. Hierfür nutzt der Social Engineer bereits eingeholte Informationen und spiegelt eine falsche Identität vor.

- Dadurch ist es ihm anschließend möglich das erhaltene Vertrauen *auszubeuten*. Das Opfer muss dazu gebracht werden, die für den Angreifer notwendigen Handlungen auszuführen und Informationen preiszugeben.

- Abschließend muss der Social Engineer die erhaltenen Daten für seine Zwecke *nutzen*. Sollte er damit seine Ziele noch nicht erreicht haben, muss er zu einer der vorigen Aktionen zurückkehren.

### *D. Schutz durch Erkennen der Einflussfaktoren*

Die Zielperson kann Angriffen von Social Engineers entgegenwirken, indem sie gewisse Einflussfaktoren beachtet. ENISA [13] zeigt in einer Checkliste vier wichtige Faktoren auf:

- Es soll die *Legitimität* der Anfrage überprüft werden und nachvollzogen werden, ob es sich um eine gewöhnliche Auskunft handelt. Die betroffene Person soll abwägen, ob sie diese Information bereitstellen darf und ob die aktuelle Situation den gewöhnlichen Weg zur Weitergabe der Daten darstellt.

- Wie bereits in vorigen Abschnitten erwähnt wurde, hat jede persönliche Information für den Angreifer einen Nutzwert. Dennoch sollte vom vermeintlichen Opfer überprüft werden, welche *Wertigkeit* die herauszugebende Information besitzt. Dies kann geschehen, indem nachvollzogen wird, in welcher Form die Information in der Öffentlichkeit missbraucht werden kann.

- Die *Herkunft* der anfragenden Partei spielt eine entscheidende Rolle. Sie soll möglichst überprüft werden, bevor vertrauliche Daten weitergegeben werden.

- *Timing* stellt den letzten Einflussfaktor dar. Zeit spielt eine wichtige Rolle bei der Entscheidungsfindung. Die befragte Partei sollte erkennen, ob sie bei der Preisgabe von Informationen unter Zeitdruck steht oder ob sie in Ruhe entscheiden kann.

Werden diese Punkte beachtet, kann ein grundlegender Schutz gegen Social Engineering Angriffe geschaffen werden.

## VII. Fallbeispiel: Social Network Sites

Social Network Sites (SNS), wie Facebook oder studiVZ, werden allgemein als ein Netz von Akteuren mit unterschiedlichen Relationen definiert, wobei die Bildung von Beziehungen durch die Funktionalität des Portals ermöglicht wird. Dabei stehen Beziehungspflege und Eigenpräsentation im Vordergrund, wobei soziale Kontakte geschaffen werden sollen [16]. Im vorangegangenen Abschnitt wurde erläutert, mit welcher ausgeklügelten Vorgehensweise Social Engineers arbeiten, um an private Informationen zu gelangen. Unter Umständen ist der Aufwand für den Angreifer jedoch auf ein Minimum beschränkt, da Menschen, angespornt durch den Einsatz moderner Medien der Social Network Sites, große Teile ihres Lebens öffentlich machen. Folglich wird ihre Privatsphäre durch eigenes Handeln oftmals massiv verletzt, welches ein erhöhtes Gefahrenpotential darstellt, Opfer von Angriffen aus der „digitalen Welt“ aber auch in der „realen Welt“ zu werden. Auf diese Weise wird durch das unbedachte Vorgehen der Individuen nicht nur die Sicherheit ihrer IT-Systeme gefährdet, sondern vielmehr wird durch die sorglose Verwendung der IT-Systeme eine Gefahrenquelle geschaffen.

### A. Psychologische Aspekte und Folgen

Die Popularität von Social Network Sites ist ungebrochen und sie erfreuen sich gerade unter jüngeren Menschen großer Beliebtheit. Gründe für den Erfolg sind in der menschlichen Verhaltensweise zu finden. Knoke [17] zeigt anhand einer Studie der ENISA [18] die elementaren psychologischen Aspekte auf:
Der Mensch ist von Natur aus nicht als Einzelgänger geschaffen. Deswegen besteht sein Bestreben darin, mit anderen Leuten in Kontakt zu stehen und Verbindungen zu knüpfen. Hierfür stellen moderne Technologien eine attraktive und neuartige Form der Kommunikation zur Verfügung, welche sich als Trend entwickelt, dem letztendlich ein beträchtlicher Anteil der Gesellschaft zu folgen weiß. Der Anreiz der Social Network Portale ist die neue Form des zwischenmenschlichen Austauschs [17], welcher Tagebuch, Kontaktbörse und Multimediaportal vereint. Mit Hilfe dieser Technik gelingt es, das Kommunikationsbedürfnis der Menschen im hohen Maße zu befriedigen.
Dabei entsteht die Hoffnung auf Anerkennung innerhalb des Netzwerks. Viele Benutzer streben das Ziel an, große Popularität zu erreichen, das die Sehnsucht nach Ansehen erfüllt. Dazu sind Menschen teilweise bereit, einen nicht geringfügigen Anteil privater Informationen preiszugeben, welcher sie nach ihrem Empfinden in einem positiven Licht erscheinen lässt. Als Gradmesser dient dazu meist die Anzahl der Einträge in der Freundesliste, weshalb Benutzer dazu verleitet werden, fremde Personen in ihren Bekanntenkreis aufzunehmen. Knight spricht hier gar von einem „Popularitätswettbewerb“ [19]. Diese unbekannten Freunde sind somit in der Lage, alle in das Portal eingestellten Daten auszulesen, um daraus Nutzen zu ziehen, wie es in Abschnitt VII-B thematisiert ist.
Darüber hinaus haben viele Nutzer keine Vorstellung über die Tragweite ihrer verbreiteten Daten, da ihnen nicht bewusst ist, wie viele Leute ihre Profile tatsächlich lesen können [17]. Diesen Aspekt untermauert eine Studie, die besagt, dass 71% der 2000 befragten Personen ihr Social Network Profil erst ihrem Arbeitgeber oder Kollegen zeigen würden, wenn sie vorher persönliche Informationen gelöscht hätten [20]. Diese Naivität resultiert aus dem Irrglauben sich unter vermeintlichen Freunden in einer sicheren, gewohnten Umgebung des Portals aufzuhalten, in dem das Gefühl von Intimität [17] entsteht.
Aus diesen Aspekten ergibt sich die Feststellung, dass der Mensch durch seinen Drang nach Ansehen und Knüpfung von neuen sozialen Kontakten bereit ist, einen großen Anteil seiner Privatsphäre aufzugeben.

### B. Gefahren

Aktivität auf Social Network Sites kann verschiedene Gefahren verstärkt hervorrufen, wie Knoke [17] darstellt:
*Stalking* bezeichnet das systematische Nachstellen einer Person. Dieser Vorgang kann durch den Missbrauch von Social Network Sites hervorgerufen oder verstärkt werden, beispielsweise durch überschwemmen des Opferprofils mit Kommentaren und Nachrichten. Mögliche Opfer ziehen eine derartige Bedrohung meist nicht in Betracht, da sie gutgläubig sind und an einen „Ehrenkodex“ innerhalb des Social Networks glauben. Christakis [21] zeigt auf, dass viele Benutzer zwar zahlreiche private Informationen veröffentlichen, sie aber dennoch als Teil der Abmachung der SNS sehen, nicht auf Schritt und Tritt verfolgt zu werden. Auf Seite des Stalkers sinkt durch die Verwendung des anonymen Online-Wegs die Hemmschwelle zur Ausführung der Tat.
*Mobbing* wird u.a. betrieben, wenn falsche, nachteilige Informationen und Gerüchte über das Opfer verbreitet werden. Durch Identitätsdiebstahl kann der Täter ein gefälschtes Social Network Profil des Opfers in Umlauf bringen, um damit die Zielperson in Verruf zu bringen. Darüber hinaus besteht die Möglichkeit über ein Profil, eines nicht existenten Individuums heraus, Kontakt mit dem Opfer aufzunehmen, welches annimmt mit der vorgetäuschten Person seine Privatsphäre zu teilen. Die bereits thematisierte intime Atmosphäre im Freundesnetzwerk verleitet die Zielperson zur Gutgläubigkeit und zweifelt nicht an der Aufrichtigkeit des Kontaktpartners. Bei psychisch labilen Menschen kann die Erkenntnis des „virtuellen Betrugs“ schwere Folgen hervorrufen, wie ein besonders dramatischer Fall [22] zeigt.
*Betriebsspionage* bezeichnet den Vorgang, den der Angreifer ausführt, um von seinem Opfer vertrauliche firmeninterne Informationen zu erlangen. Auch hier zeigt sich die Zielperson innerhalb einer Online-Gemeinschaft meist kooperativ, da der Angreifer als vertrauenswürdige Onlinebekanntschaft eingestuft ist und somit unter vermeintlichen Freunden eine Bereitschaft existiert, sensible Daten ohne weitere Sicherheitsbedenken herauszugeben. Auch der Aspekt des Geltungsbedürfnisses kann hier mit einbezogen werden. Durch das Bereitstellen von Insider-Informationen kann die Zielperson das Gefühl entwickeln, sich innerhalb der Community profilieren zu können.

Als *Spam* werden unerwünschte Mitteilungen oder Nachrichten bezeichnet, die der Zielperson unaufgefordert zugesandt werden. In diese Kategorie fällt beispielsweise der Versand von Emails mit Werbungsinhalt, aber auch die in Abschnitt VI-B dargestellten Phishing Mails. Um diese Nachrichten effizienter für die entsprechenden Zielpersonen zuzuschneiden, analysiert der Angreifer die einzelnen Social Network Profile der Opfer um personalisierten Spam zu erstellen. Zum Beispiel kann durch das Auslesen der Hobbies einer Zielperson auf Konsumgüter geschlossen werden, welche sie unter Umständen attraktiv findet oder sie mit ihrem vollständigen Namen angesprochen werden. Darüber hinaus können Spam oder Phishing Mails direkt in den Social Network Portalen versandt werden, so dass die Zielperson nicht nur beeinflusst durch den Inhalt (u.a. Anrede, persönlich zugeschnittener Text) sondern auch durch den Ursprung der Nachricht an ihre Vertrauenswürdigkeit glaubt. Die Sicherheitsfirma Trusted Defender Labs stellt fest, dass bei gewöhnlichen Phishing Attacken der Angreifer ungefähr 5% nutzbare Antworten erhält. Wird jedoch das Mittel der personalisierten Nachricht eingesetzt, steigt die Erfolgsrate auf bis zu 80% [23].

### C. Datenschutzprobleme

Wie im vorangegangenen Abschnitten erläutert wurde, ist der Nutzer von Social Network Sites einer Menge von Gefahren ausgesetzt. Doch auch wenn dieser mit Umsichtigkeit und Vorsicht jene Portale verwendet, um die erwähnten Probleme zu minimieren, existieren dennoch weitere Gefahren, welche sich kaum ausgrenzen lassen.

Der Handel mit persönlichen Daten ist ein lukratives Geschäft, so dass es eine Vielzahl an technischen Möglichkeiten gibt, diese einzuholen und aufzubereiten. ENISA [18] weist darauf hin, dass es beinahe uneingeschränkt möglich ist, die Profildaten der Social Network Nutzer automatisch auszulesen und zu speichern. Darüber hinaus können Fotos aus Profilen oder Fotoalben mit Bilderkennungssoftware analysiert werden, damit in einem zweiten Schritt die betreffenden Personen auf anonym eingestellten Fotos wiedererkannt werden können. Ist der Nutzer eines Social Networks auf diese Weise erfasst worden, so kann dieser beispielsweise in einem anonymen Dating Forum erkannt werden.

Einen weiteren technischen Aspekt betrifft das SN Portal selbst. ENISA [18] zeigt auf, dass es Nutzern kaum möglich ist, ihren Account vollständig zu löschen und alle erstellten Informationen über ihre Person zu entfernen. Hier bleiben oftmals persönliche Daten zurück, etwa in den Kommentarfeldern anderer Profile.

Auch bei umsichtigen Nutzern sorgen nicht nur technische Aspekte für ein gesteigertes Datenschutzproblem. Knoke [17] beschreibt, dass „allzu kontaktfreudige Freunde“ Daten und Bilder über die Zielperson im Netz verbreiten, auch wenn diese selbst achtgibt, kaum Spuren im Netzwerk zu hinterlassen. So werden beispielsweise die Fotos der letzten Party ohne Rückfrage eingestellt und Markierungen mit Links und Namen erstellt.

Es zeigt sich, dass Social Network Sites unter einem hohen Datenschutzproblem leiden, da sie sowohl dem Social Engineer eine Plattform bieten, auf einfache Weise an private Informationen zu gelangen, als auch den naiven Benutzer herausforderen, seine Privatsphäre einzuschränken.

## VIII. Zusammenfassung und Ausblick

IT-Sicherheit hängt von verschiedenen Faktoren ab. Der technische Aspekt spielt dabei eine wichtige Rolle, ist jedoch von den Einflüssen des menschlichen Handelns abhängig. Der Mensch als IT-Benutzer fungiert somit als Basis jeglicher IT-Sicherheit. Da er durch psychologische Faktoren leicht beeinflussbar ist, stellt dieser das schwächste Glied in der Sicherheitskette dar.

Es wurde aufgezeigt, dass Individuen oftmals keiner rationalen Handlungsweise folgen, sondern sich auf Grund von verschiedenen situationsbedingten Einflüssen leiten lassen. Die Erwartungstheorie und der Besitztumseffekt zeigen dies unter anderem auf. Somit treffen sie die Entscheidung über die Herausgabe von sensiblen Daten nach subjektivem Empfinden, wobei sie zwischen Gewinn und Risiko abzuwägen haben. Da an dieser Stelle unvorhersehbare Faktoren und fehlerhafte Einschätzungen auftreten, wird der Schutz der Privatsphäre riskiert, wobei durch das Bereitstellen persönlicher Daten die IT-Sicherheit in Gefahr gerät.

Diese Schwachstelle kann ein Angreifer für seine Zwecke verwenden, um etwa mit dem Prinzip des Social Engineerings das menschliche Verhalten auszunutzen, um an vertrauliche Informationen zu gelangen. Dies geschieht meist durch fingierte Anrufe des Angreifers, der durch die geschickte Anwendung von psychologischen Erkenntnissen sein Opfer manipuliert.

Im Fallbeispiel der Social Network Sites wurde dargestellt, dass Menschen bereits ohne die Bedrängnis von trickreichen Angreifern bereit sind, einen großen Teil ihrer Privatsphäre aufzugeben, bedingt durch das Verlangen nach Ansehen und zwischenmenschlichen Kontakten. Dabei konnte festgestellt werden, dass IT-Sicherheit aus zwei Blickwinkeln zu sehen ist: IT-Systeme werden nicht nur durch die Offenlegung von vertraulichen Informationen gefährdet, sondern IT-Systeme bedrohen auch durch die sorglose Nutzung der Menschen ihre Privatsphäre.

Aus diesen Gründen muss der Benutzer eine Awareness entwickeln, die sein Sicherheitsbewusstsein sensibilisiert, damit vertrauliche Daten geschützt und die IT-Sicherheit bewahrt werden kann.

## Literatur


[1] B. Esslinger, “Die drei Säulen der IT-Sicherheit,” April 2005.

[2] W. Degenhardt, “IT-Sicherheit - Recht, Wirtschaft und Kultur,” Oktober 2008.

[3] W. Degenhardt, “Psychologie der IT-Sicherheit,” 2009. [Online]. Available: http://www.silicon.de/mittelstand/0,39038986,41001030-2,00/psychologie+der+it_sicherheit_+teil+2.htm

[4] W. Degenhardt, “Psychologie der IT-Sicherheit,” 2009. [Online]. Available: http://www.silicon.de/mittelstand/0,39038986,41001030,00/psychologie+der+it_sicherheit_+teil+2.htm

[5] P. D. L. Prechelt, “Vorlesung Anwendungssysteme - Privatsphäre.”

[6] T. Weichert, “Privacy and Human Behaviour,” Januar 2009.

[7] A. Acquisti, “What can Behavioral Economics Teach Us About Privacy?” 2007.
[8] G. Kirchgässner, *Homo oeconomicus: Das ökonomische Modell individuellen Verhaltens und seine Anwendung in den Wirtschafts- und Sozialwissenschaften*. Mohr Siebeck, 2000.
[9] K. Müller, M. Rosenthal, K. Reins, and A. Miller, “Die Rationalität ökonomisch handelnder Akteure,” 2006.
[10] B. Schneier, “How to Sell Security,” 2008. [Online]. Available: http://www.cio.com/article/367913/How_to_Sell_Security?page=1
[11] D. Kahneman, J. L. Knetsch, and R. H. Thaler, “Anomalies: The Endowment Effect, Loss Aversion, and Status Quo Bias,” 1991.
[12] A. Zeichick, “Zeichick’s Take: The Endowment Effect meets software development,” 2009. [Online]. Available: http://www.sdtimes.com/SearchResult/33358
[13] Maria Papadaki and Steve Furnell and Ronald C. Dodge, “Social engineering: Exploiting the weakest links,” 2008.
[14] C. Grafe, “Seminar Datensicherheit: Social Engineering - über die Gefahren der psychologischen Verwirrung bei der Informationsbeschaffung und die Sicherheitskultur,” 2007.
[15] K. Mitnick and W. Simon, *Die Kunst der Täuschung - Risikofaktor Mensch*. mitp, 2002.
[16] G. W. Loub, L. Weber, and P. M. Bobrowsky, “SNS - mehr als nur virtuell?” 2008. [Online]. Available: http://www.unet.univie.ac.at/~a9000165/php/mume_bobrowsky/?page_id=37
[17] F. Knoke, “Die Gefahren des sozialen Netzes,” 2007. [Online]. Available: http://www.spiegel.de/netzwelt/web/0,1518,517584,00.html
[18] G. Hogben, “Security Issues and Recommendations for Online Social Networks,” Oktober 2007.
[19] M. Knight, “Real friends and virtual strangers,” 2008. [Online]. Available: http://www.cnn.com/2007/TECH/09/13/fbook.friends/index.html
[20] A. Schütz, “Social Networks als Karrierekiller,” November 2007. [Online]. Available: http://www.silicon.de/cio/strategie/0,39038989,39160370,00/social+networks+als+karrierekiller.htm
[21] S. Weingarten, “Selbstentblößung im Internet - Tiefes menschliches Bedürfnis,” 2008. [Online]. Available: http://www.spiegel.de/unispiegel/wunderbar/0,1518,547445,00.html
[22] F. Patalong, “Cyber-Mobbing - Tod eines Teenagers,” 2007. [Online]. Available: http://www.spiegel.de/netzwelt/web/0,1518,518042,00.html
[23] J. Brien, “MySpace und Co öffnen Phishern Tür und Tor,” 2007. [Online]. Available: http://pressetext.de/news/071002020/myspace-und-co-oeffnen-phishern-tuer-und-tor/

# Tor und Angriffe gegen Tor

Marc Ströbel
Betreuer: Heiko Niedermayer
Seminar Innovative Internettechnologien und Mobilkommunikation SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: stroebem@in.tum.de

***Kurzfassung*— Viele Internetnutzer vertrauen auf den Dienst der Anonymisierungssoftware Tor. Die von Tor bereitgestellte Anonymität beruht auf der Anwendung einiger informationstechnischer Sicherheitskonzepte. In dieser Ausarbeitung wird detailliert die Funktionsweise des Tor-Protokolls, sowie existierende Schwächen im Tor-Design aufgezeigt.**

***Schlüsselworte*— Tor, Anonymität, Onion Routing, Privacy, Datenschutz**

## I. Einleitung

Es bestehen vielfältige Gründe für den Wunsch nach Anonymität im Internet. Motive können zum Beispiel sein: freie Meinungsäußerung in Ländern mit strengen Gesetzen, das anonyme Betreiben eines gesellschafts- oder politikkritischen Blogs oder einfach das Anliegen unerkannt Daten auszutauschen. Das Tor-Projekt steht im Kontext des bereits 1996 von Goldschlag, Reed und Syverson vorgestellten Verfahrens des Onion Routings, welches durch den Einsatz von Proxy-Servern die Quelle des Nachrichten-Austausches in Anwendungsprotokollen, wie http oder telnet, anonymisiert [1]. Leider existieren im Onion Routing Protokoll signifikante Schwächen, wie die fehlende *perfect forward secrecy*, das Fluten von Knoteninformationen über das Netzwerk, sowie die fehlende Integritätskontrolle der Daten. Tor als nächste Generation des Onion Routings, eingeführt von Dingledin, Mathewson und Syverson, beseitigt diese Schwächen und hat neben der Einführung von versteckten Diensten vor allem die Benutzerfreundlichkeit im Fokus [2].
Der nächste Abschnitt beschäftigt sich mit der Grundidee des Onion Routings, bevor Abschnitt 3 Tor im Detail vorstellen wird. Abschnitt 4 diskutiert grundsätzlich mögliche, sowie einige ausgewählte, Angriffe auf Tor. Abschließend gibt Abschnitt 5 eine kurze Zusammenfassung, sowie einen Ausblick auf die weitere Entwicklung des Tor-Projekts.

## II. Onion Routing

Beim Onion Routing durchlaufen Nachrichten einen Pfad von Netzwerkknoten, in welchem jeder Knoten nur seine unmittelbaren Vorgänger und Nachfolger kennt. Folglich ist der Sender ausschließlich dem ersten Knoten und der Empfänger ausschließlich dem letzten Knoten und dem Sender bekannt. Dies wird durch ein umfangreiches Protokoll sowie einer Public-Key-Infrastruktur sichergestellt.

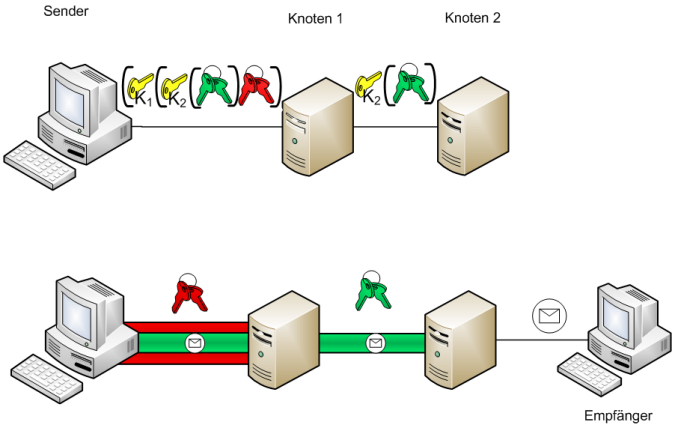



Abbildung 1. Onion Routing

Vereinfacht lässt sich die Idee wie folgt darstellen: Der Sender ist in Besitz der öffentlichen Schlüssel der Knoten ($K_1$ bis $K_n$) und kann somit in einem initialen Verbindungsaufbau sicher mit jedem Knoten zwei gemeinsame Schlüsselpaare austauschen. Danach kann jede Nachricht in $n$ Schichten verschlüsselt werden und beim Versenden über den Pfade sukzessiv von jeden einzelnen Knoten entschlüsselt zu werden (siehe Abbildung 1). Diese Struktur erinnert an die Eigenschaft einer Zwiebel, daher trägt der englische Name auch das Wort Onion im Titel, welches ins Deutsche übersetzt Zwiebel bedeutet. Durch diese Umhüllung der Nachricht ist nur der letzte Knoten des Pfades in der Lage die Adresse des Empfängers zu entschlüsseln. Des Weiteren spezifiziert das Protokoll den Aufbau eines solchen Pfades, sowie einige Maßnahmen zum Schutz gegen Wiedereinspielungen und zur Einschränkung einer passiven traffic Analyse. Diese Pfade werden im Folgenden auch gleichbedeutend *Circuit* genannt.

## III. Tor - the Second Generation Onion Routing

Als zweite Generation des Onion Routings, basiert Tor auf der Idee von Goldschlag, Reed und Syverson, verbessert jedoch einige sicherheitskritische Eigenschaften und legt Wert auf den einfachen und flexiblen Einsatz des Protokolls [2].
Der flexible Einsatz ist bei den heutigen Anwendungen, die oftmals in Echtzeit Dienste zu Verfügung stellen, stark von geringen Latenzzeiten abhängig, daher wurde Tor als *Low-*

*Latency Anonymous Network* konzipiert. Somit besitzt Tor im Vergleich zum Onion Routing entschieden unterschiedliche Anforderungen.
Tor implementiert zudem *perfect forward secrecy*, welches bedeutet dass die einzelnen Schlüssel eines Circuits, im Gegensatz zum Onion Routing, nicht aufeinander basieren. Daher macht das erfolgreiche knacken eines Schlüssels an einem Zwischenknoten kein Entschlüsseln der Daten an Folgeknoten möglich. Gegen willkürliche Veränderungen der versendeten Pakete wurde eine End-zu-End Datenintegritätskontrolle eingeführt.
Zusätzlich enthält das Tor-Design *Directory Server* zur Verbreitung von Informationen über Tor-Knoten und durch *Rendezvous points* die Möglichkeit von versteckten Diensten.

### A. Die Architektur von Tor

Im Wesentlichen besteht das Tor-Netzwerk aus drei Komponenten:

- dem *Onion Proxy*, welcher lokal auf dem Computer des Benutzers läuft. Er stellt einen SOCKS Proxy [3] für ausgehende TCP Verbindungen bereit und verteilt diese auf von ihm initiierten Pfaden bzw. Circuits.
- den *Onion Routern* (im Folgenden auch OR), sie dienen als Netzwerkknoten und gehen daher Circuit Verbindungen zu anderen Onion Routern oder Onion Proxys ein und führen ggf. die gewünschte Anfrage an den eigentlichen Kommunikationspartner des Clients aus.
- den *Directory Servers*, eine kleine Anzahl an vertrauenswürdigen Servern, welche eine signierte Liste aktuell bekannter Onion Router, die deren Deskriptoren enthält, bereitstellen. Ihre öffentlichen Schlüssel sind vorab im Tor Quellcode hinterlegt.
  Die Deskriptoren führen aktuelle Status Informationen und die öffentlichen Teile des Identitätsschlüssels und des kurzlebigen *Onion Keys*. [4]

Die Verbindungen zwischen den einzelnen Komponenten sind durch das TLS Protokoll [5] gegen unerwünschte Modifikation geschützt und verschlüsselt. Jeder Onion Router authentifiziert sich anhand Zertifikaten, die mit seinem Identitätsschlüssel signiert wurden.

### B. Das Tor-Protokoll im Detail

Grundsätzlich unterscheiden Onion Router zwei verschiedene Nachrichtentypen:

*1) Control Cells:* Control Cells dienen hauptsächlich zur Steuerung des Circuits und werden direkt von dem Router der sie empfängt ausgeführt. In der Abbildung 2 wird der Aufbau einer Control Cell dargestellt. Das Feld *CircID* enthält eine Circuit ID, welche im folgenden noch erläutert wird. Die Felder *CMD* und *DATA* beinhaltet den auszuführenden Befehl und dessen Kontextinformationen, wie beispielsweise bei einem Aufbau eines Circuits die Adresse des ersten Onion Routers [2].

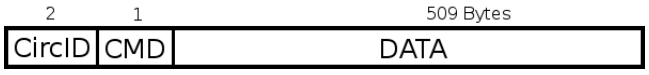



Abbildung 2. Der Aufbau einer Control Cell

*2) Relay Cells:* Relay Cells wandern den Circuit entlang und enthalten meist End-zu-End Verbindungsdaten oder Control Cells für einen nachfolgenden Knoten, für beide Arten gilt, dass nur letztendliche Empfänger das Daten Feld entschlüsseln kann. Im Falle von End-zu-End Verbindungsdaten ist das der letzte Onion Router im bestehenden Circuit. Bei der Übertragung von Control Cells ist das der Knoten an welchen der Befehl gerichtet ist.
Das Feld *Relay* enthält den Relay Typ, beispielsweise *relay data* zum Austausch von Verbindungsdaten oder *relay begin* für das öffnen eines Streams. Weitere Felder sind die *StreamID*, welche zur eindeutigen Identifizierung des Streams innerhalb eines Circuits dient, sowie das Feld *Digest*, das in die Integritätskontrolle einfließt. Der Wert von *Len* gibt die Länge des Datenfeldes an [2].

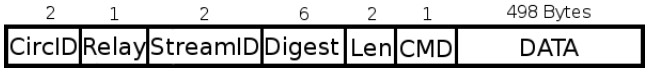



Abbildung 3. Der Aufbau einer Relay Cell

Betrachten wir im folgenden Abschnitt den Aufruf einer Website unter Verwendung des Onion Proxys (siehe Abbildung 4):

*3) Die Erzeugung eines Circuits:* Zunächst erhält der Onion Proxy des Benutzers auf Anfrage die Deskriptoren einiger Onion Router. Aus dieser Menge wählt er zufällig, in Abhängigkeit seiner Konfiguration, einen Einstiegspunkt für seinen Circuit aus. Nach dem Aufbau der TLS Verbindung sendet er diesem eine *create* Control Cell mit der CircID c1 und der ersten Hälfte eines Diffie-Hellman[1] Handshakes $g^{x_1}$, RSA verschlüsselt mit dem Onion Key des Routers. Der Onion Router antwortet dem Proxy mit der Control Cell *created* und sendet ihm neben der CircID und seinen Teil des Diffie-Hellman Handshakes $g^{y_1}$, einen Hash über den gemeinsam errechneten Schlüssel $K_1$ (Nach dem Diffie-Hellman Verfahren gilt: $K_1 = g^{x_1 y_1}$ [6, Kap. 8]).
Das bestimmen einer CircID dient ausschließlich der eindeutigen Identifikation des Circuits zwischen Onion Proxy und Onion Router. Diese Maßnahme beugt dem Umstand, dass zwei Netzwerkknoten simultan in mehreren Circuits benachbart sein können vor.
Nach dem erfolgreichen Austausch des symmetrischen Schlüssels $K_1$ kann der Onion Proxy Relay Cells mit dem Onion Router austauschen. Diese sind verschlüsselt im AES. Auf Wunsch kann der sichere Pfad um zusätzliche verfügbare

[1]Das Diffie-Hellman Verfahren dient im Allgemeinen zum Aufbau eines sicheren Kanals, ohne bereits vorab ausgetauschte Schlüssel zu verwenden.

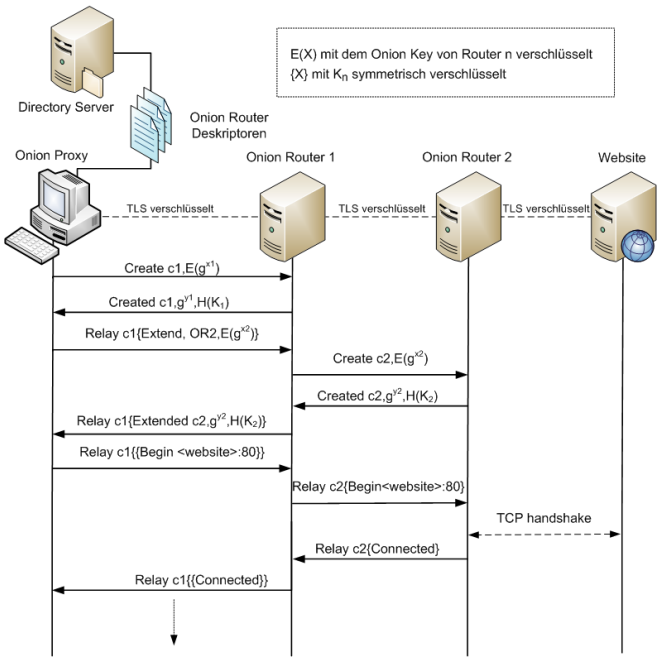



Abbildung 4. Das Tor Protokoll im Detail

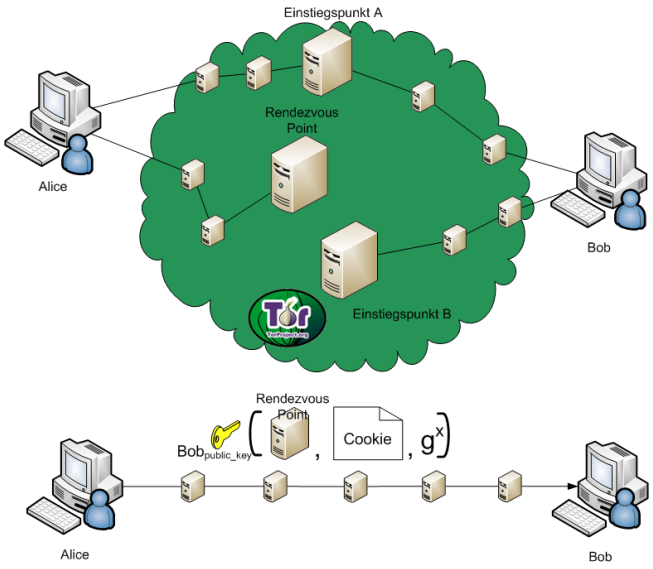



Abbildung 5. Verbindungsaufbau zu einem versteckten Dienst: 1. Schritt

Knoten erweitert werden. Dazu sendet der Onion Proxy die *extend* Relay Cell: $Relay\ \ c1(Extend, OR2, E(g^{x_2}))$, welche die Adresse des Erweiterungknotens OR2 angibt. Der Schlüsselaustausch läuft analog zum *create* Befehl ab.

*4) Der Austausch von Verbindungsdaten:* Hat ein Benutzer nun einen Circuit mit Onion Routern aufgebaut, ist er in der Lage über diesen jegliche TCP basierte Kommunikation zu tunneln. Dazu dienen unter anderem die Relay Cells *begin* und *data*. *Begin* öffnet einen neuen Stream zu einem im Befehl definierten Ziel – *data* überträgt Daten in beide Richtungen.
Durch die vorab ausgetauschten Diffie-Hellman Schlüssel wird entsprechend dem Onion Routings sichergestellt, dass alle Knoten des Pfades durchlaufen werden und nur der berechtigte Onion Router die Adresse des Kommunikationspartners oder des neu hinzuzufügenden Onion Routers kennt. Der letzte Knoten im Circuit tritt daher für den Empfänger der Daten wie der eigentliche Sender auf.
Durch die bereits erwähnte Integritätsüberprüfung, ist sichergestellt, dass die Pakete zwischen den Enden des Pfades nicht kompromittiert werden. Sie basiert auf Hashes der versendeten Daten, der Datenhistorie und einem Schlüssel-Derivat zwischen Sender und den letzten Router des Circuits [2].

### C. Rendezvous Points - Tors Hidden Services

Wie der Name vermuten lässt, handelt es sich bei den Hidden Services um das anonyme Anbieten von Diensten, wie beispielsweise Webseiten oder eines *Anonymus Upload Centers*. Das ursprüngliche Onion Routing zog bereits eine komplett anonyme Kommunikation, ähnlich dem Rendezvous Points Prinzip, in Betracht, blieb jedoch die konkrete Spezifikation schuldig. [1]

*1) Das Anbieten eines Hidden Service:* Der Onion Proxy des Benutzers wählt einige Onion Router als Einstiegspunkte für den versteckten Dienst aus. Die Adressen dieser Router werden, signiert vom Benutzer, anonym durch den Onion Proxy veröffentlicht. Danach sind sie über Tor durch die virtuelle Domain $z.onion$ erreichbar. Wobei $z$ einen Hash des öffentlichen Schlüssel des Benutzers darstellt. [7]

*2) Verbindung zu einem Hidden Service herstellen:* Angenommen die Benutzerin Alice möchte auf einen Hidden Service des Benutzers Bob zugreifen (Zur Vereinfachung repräsentieren Alice und Bob ihre jeweiligen Onion Proxys): Nachdem Alice von Bobs Dienst gehört hat und darauf zugreifen will, wählt sie einen Onion Router als ihren Rendezvous Point aus. Sie bildet einen Circuit zu diesem Onion Router und schickt Bob, über einen seiner Einstiegspunkte, eine Nachricht. Diese Nachricht, verschlüsselt mit Bobs öffentlichen Schlüssel, enthält die Adresse ihre Rendezvous Point, ein zufällig erstelltes *rendezvous* Cookie und die erste Hälfte eines Diffie-Hellman Handshakes (siehe Abbildung 5). Möchte nun Bob den Dienst für Alice anbieten, bildet er einen Circuit zu ihrem Rendezvous Point (mit der *establish rendezvous* Relay Cell) mit dem erhaltenen Cookie und Bobs Hälfte des Diffie-Hellman Handshakes (siehe Abbildung 6).
Nun sind die beiden Benutzer durch Alices Rendezvous Point verbunden und können ganz normal der Relay Cell *begin* einen Datenaustausch beginnen [2].

## IV. Angriffe gegen Tor

In diesem Abschnitt werden zunächst einige grundsätzlich mögliche Angriffe, welche bereits von den Entwicklern des

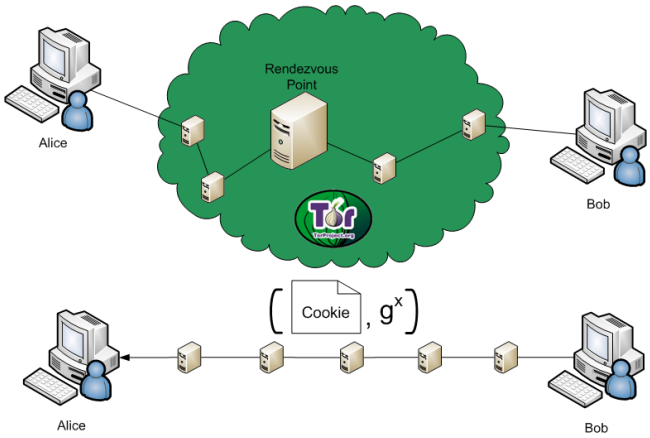



Abbildung 6. Verbindungsaufbau zu einem versteckten Dienst: 2. Schritt

Tor-Protokolls diskutiert wurden [2], betrachtet. Darüber hinaus werden im zweiten Teil zusätzlich ausgewählte Angriffe gegen Tor aus weiteren Publikationen vorgestellt.

### A. Grundsätzlich Mögliche Angriffe

#### 1) Passive Angriffe:

*a) Auslesen der User Daten:* Es ist einem Angreifer möglich die unverschlüsselten Daten, die am Ende des Circuits zum eigentlichen Kommunikationspartner des Benutzer übertragen werden, auszulesen. Es besteht zusätzlich die Möglichkeit, dass der Benutzer über den Inhalt dieser Daten seine eigentliche Identität preisgibt. Als Gegenmaßnahme schlagen Dingledin, Mathewson und Syverson die Verwendung von Webfiltern wie Privoxy vor, welche Applikationsdaten filtern und verräterische Muster entfernen [2]. Ein weiterer Ansatz für die Informationsvertraulichkeit ist die Verwendung von Verschlüsselung in höheren Netzwerkschichten wie beispielsweise durch das Protokoll SSL/TLS.

*b) Verkehrsflussanalyse:* Angenommen der Angreifer kann sowohl den Datenverkehr des eigentlichen Senders als auch des Empfängers überwachen, so ist er in der Lage durch Zeit- oder Datengröße-Korrelationen diese Benutzer bzw. Maschinen genau zu bestimmen. Auch die Tatsache, dass mehrere TCP Verbindungen simultan einen Circuit benutzen können, gibt nur wenig Schutz gegen diese Analysen.
Schutz gegen diese Art von Angriffen kann durch das Verstecken der Verbindung zwischen dem Onion Proxy und den ersten Onion Router gegeben werden. Um dies zu erreichen, ist es denkbar als Benutzer selbst einen Onion Router zu betreiben und sich grundsätzlich lokal zu diesem, als ersten Knoten im Circuit, zu verbinden. [2]
Auch gegen das *Website fingerprinting*, das im Grunde eine Art der Verkehrsflussanalyse ist, besitzt Tor eine Anfälligkeit. Die Idee hinter diesem Angriff ist nicht nur Sender und Empfänger zu belauschen, sondern ein Verkehrsmuster bestimmter Webseiten zu erstellen und diese im Tor Datenverkehr wiedererkennen.

#### 2) Aktive Angriffe:

*a) Den Onion Proxy kompromittieren:* Ein kompromittierter Onion Proxy, der beispielsweise zu jeder aufgebauten TCP Verbindung seine und die Identität des Empfängers dem Angreifer preisgibt, besitzt durch das Einführen von versteckten Features eine ähnliche Charakteristik wie ein Trojaner [6, Kap.2]. Um dieser Bedrohung vorzubeugen veröffentlicht das Tor-Projekt Hashwerte ihrer aktuellen Versionen.

*b) Eine DoS Attacke gegen unbeobachtete Onion Router:* Der Erfolg einer Verkehrsflussanalyse hängt maßgeblich von der Anzahl der beobachteten Onion Router ab. Daher bietet es sich als Angreifer an, diese Anzahl gezielt durch eine Denial-of-Service Attacke zu verringern. Insbesondere, da die Adressen der verfügbaren Tor-Knoten von den Directory Servern abrufbar sind.

*c) Einen bösartigen Onion Router betreiben:* Das Betreiben eines Onion Routers kann ausschließlich in dem Fall, dass dieser sogleich der erste als auch der letzte Knoten im Circuit ist, Informationen über den Benutzer preisgeben. Trifft nur eine der beiden Bedingungen zu, kann der Angreifer keine Zuordnung zwischen den gesendeten Daten und den aktiven Onion Proxys vornehmen. Die Kontrolle eines Onion Routers in der Mitte eines Circuits ist aufgrund der Unkenntnis der Schlüssel zwischen dem Benutzer und den äußeren Knoten nutzlos. Durch das Betreiben vieler solcher Onion Router, steigt jedoch die Chance auf Erfolg rapide an.

*d) Angriffe auf Directory Server:* Aufgrund der besonderen Stellung der Directory Server in der Tor Architektur, existieren mögliche Szenarien mit drastischen Auswirkungen auf die Sicherheit der Anonymisierungssoftware. Das Ziel eines Angreifers könnte es sein, die Adressen seiner kompromittierten Onion Router über die Directory Server zu verteilen. Jedoch ist Tor so konzipiert, dass grundsätzlich nur Onion Router angeboten werden, welche von der Mehrheit der Directory Server unterstützt werden [2]. Trotz dieser Maßnahme ist Robustheit der Directory Server sehr wichtig, da natürlich nicht nur bösartige Knoten hinzugefügt, sondern auch intakten Knoten entfernt werden könnten, so dass nach der Übernahme von genügend Directory Servern, das Tor Netzwerk nur noch aus feindlichen Onion Routern bestehen würde.

### B. Ausgewählte Angriffe gegen Tor

*1) Low Cost Traffic Analysis of Tor:* Murdoch und Danezis haben gezeigt [8], dass eine Verkehrsflussanalyse gegen Tor, nicht ausschließlich von einem globalen Angreifer, der Sender und Empfänger gleichzeitig überwachen kann, durchführbar ist. Ihr Vorgehen nutzt die geringe Latenzzeit von Tor aus, in welcher begründet liegt, dass Relay Cells weder bewusst verzögert, neu geordnet oder sonstig zur Abwehr von Verkehrsflussanalysen besonders behandelt werden. Die Tor-Knoten wenden ausschließlich eine Art *Round-Robin* an um den mit ihnen verbundenen Circuits Fairness zu gewährleisten. Dieses Verhalten verursacht, dass die Auslastung auf dem Onion Router, die Latenz auf allen laufenden Circuits gleichermaßen beeinträchtigt. Dies machen sich Murdoch und Danezis in folgendem Angriffsmodell zu Nutze:

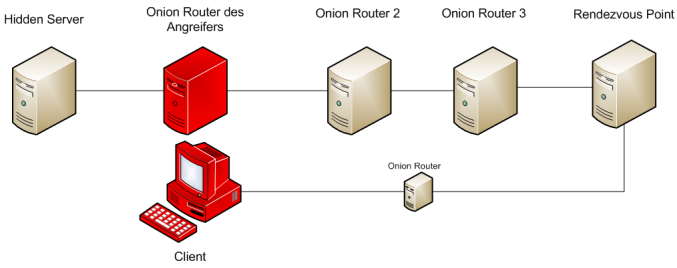



Abbildung 7. Locating Hidden Servers

Der Angreifer besitzt einen kompromittierten Server, auf welchen der anonyme Benutzer der Tor-Verbindung zugreifen möchte. Zusätzlich verfügt er über einen, aus der Sicht von Tor vertrauenswürdigen, nicht gesperrten, Onion Router. Nun erzeugt der Server ein bestimmtes Daten-Muster, beispielsweise immer wieder einen kurzen Ausbruch an Daten und dann wieder nichts. In den kurzen Phasen der Belastung kann nun der Onion Router des Angreifers durch *pingen* die Latenz der anderen Onion Router im System messen. Wenn bei mehreren Belastungsphasen der gleiche Router eine schlechte Latenz besitzt ist er Teil des vom Benutzer verwendeten Circuits [8]. Der Angreifer ist auf die Verwendung eines eigenen Onion Router angewiesen, da es sein könnte, dass manche Tor-Knoten nur auf Anfragen anderer Knoten reagieren und nicht auf Computer außerhalb des Overlay Netzwerks.
Durch dieses Verfahren ist es dem Angreifer möglich den Circuit des Tor-Benutzers Stück für Stück zurückverfolgen und die von Tor bereitgestellte Anonymität auf die eines *normalen* Proxy Servers zu verringern.

*2) Angriffe gegen Hidden Services:*

*a) Locating Hidden Servers:* Im Angriffszenario von Øverlier und Syverson [9] auf Hidden Services, besitzt der Angreifer wiederum einen Onion Router und einen Client mit leicht modifizierten Onion Proxy um auf einen angebotenen Hidden Service zuzugreifen. Das Ziel des Angreifers ist es, seinen Onion Router als ersten Knoten in den Circuit des Hidden Servers zum gemeinsamen Rendezvous Point zu platzieren (siehe Abbildung 7). Gelingt ihm dies kann er direkt die IP des Hidden Server einsehen.
Ob sich der Onion Router des Angreifers in direkter Verbindung mit dem Hidden Service befindet, ermittelt er durch bestimmte Muster, welche der Client über den Rendezvous Punkt an den Hidden Service sendet. Auch die Client Modifikation, dass dieser sich direkt mit dem Rendezvous Point verbindet, dient diesem Zweck. Zusätzlich trennt sich der Onion Proxy nach jedem erfolgreich übertragenen Muster vom Hidden Service, um einen Neuaufbau der Circuits zu erzwingen. Durch diese Änderung wird die benötigte Zeit bis der Onion Router des Angreifers in den Circuit des Hidden Servers kommt drastisch verringert. Øverlier und Syverson zeigten dass in einem Tor-Netzwerk mit etwa 250 Knoten die Lokalisierung des Hidden Services zwischen drei und 28 Minuten dauerte [9].

*b) Revealing Hidden Services by their Clock Skew:* Hohe Prozessorauslastung, welche auch bei häufiger Ver- und Entschlüsselung von Daten wie im Tor-Netzwerk auftritt, beeinträchtigt die Genauigkeit der Uhrzeit eines Systems. Steven Murdoch benutzt diese Ungenauigkeit zur Enttarnung von Hidden Services, indem er den Unterschied ob ein potenzieller Hidden Server aktiv oder inaktiv ist anhand von TCP Timestamps ausliest [10].
So kann der Angreifer durch einen Client Daten mit dem Hidden Service austauschen und währenddessen mit einer anderen Maschine Timestamps von potenziellen Kandidaten für den Hidden Server anfordern. Anhand dieser Timestamps lässt sich die Aussage treffen ob der mögliche Kandidat der Hidden Server ist oder nicht.

## V. Zusammenfassung und Ausblick

Das Tor-Projekt insgesamt ist trotz Schwachstellen, wie der von Murdoch und Danezis aufgezeigten Möglichkeit einer effektiv durchführbaren Verkehrsflussanalyse oder Øverliers und Syversons Angriff auf die Anonymität eines Hidden Services, zweifellos ein großer Schritt für die Anonymität und Privatsphäre im Internet.
Interessanterweise entpuppt sich die größte Stärke Tors, die geringe Latenz, auch als größte Schwäche des Overlay Networks, da diese die Verkehrsflussanalyse der einzelnen Streams ermöglicht, welche von allen drei ausgewählten Bedrohungen genutzt wird. Der allbekannte Kompromiss zwischen Anonymität und Benutzerfreundlichkeit scheint, wenn auch nur im geringen Maße, Tor einzuholen.
Dessen ungeachtet lohnt sich ein Blick auf die nahe Zukunft des Onion Routing Projektes: In einer Roadmap [11] für die Jahre von 2008 bis 2011 steht neben Performanz steigernden Verbesserungen, zum Beispiel der Verbindungsstabilität des Windows XP und Vista Onion Proxys, auch die Zuverlässigkeit und Sicherheit der Hidden Services auf dem Programm der Entwickler. Gelingt es dem Tor-Projekt kontinuierliche seine Netzwerkperformanz zu verbessern und einen Weg gegen die passive Verkehrsanalyse zu finden, dürfte die Technik des Onion Routings endgültig ausgereift sein und weiterhin wachsend eingesetzt werden.

## Literatur

[1] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, “Hiding Routing Information,” in *Proceedings of Information Hiding: First International Workshop*, R. Anderson, Ed. Springer-Verlag, LNCS 1174, May 1996, pp. 137–150.
[2] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router,” in *Proceedings of the 13th USENIX Security Symposium*, August 2004.
[3] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones, “Socks protocol version 5,” RFC 1928 (Proposed Standard), Internet Engineering Task Force, March 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1928.txt
[4] “Tor directory protocol version 3 specification,” August 2006. [Online]. Available: http://www.torproject.org/svn/trunk/doc/spec/dir-spec.txt
[5] T. Dierks and E. Rescorla, “The transport layer security (tls) protocol version 1.2,” RFC 5246 (Proposed Standard), Internet Engineering Task Force, August 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt
[6] C. Eckert, *IT-Sicherheit : Konzepte - Verfahren - Protokolle*, studienausg ed., Oldenbourg, 2001.

[7] "Tor rendezvous specification," August 2008. [Online]. Available: http://www.torproject.org/svn/trunk/doc/spec/rend-spec.txt
[8] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*. IEEE CS, May 2005.
[9] L. Øverlier and P. Syverson, "Locating hidden servers," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE CS, May 2006.
[10] S. J. Murdoch, "Hot or not: Revealing hidden services by their clock skew," in *Proceedings of CCS 2006*, October 2006.
[11] R. Dingledine, "Tor development roadmap, 2008-2011," December 2008. [Online]. Available: https://git.torproject.org/checkout/tor/master/doc/roadmaps/2008-12-19-roadmap-full.pdf

# Patente- Einführung, wirtschaftliche Bedeutung von Patenten auf Kommunikationsprotokolle

Hanxi Liu
Betreuer: Heiko Niedermayer
Seminar Innovative Internettechnologien und Mobilkommunikation SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Institut für Informatik
Technische Universität München
hanxi.liu@mytum.de

## Kurzfassung
Das Wort „Patent“ ist uns schon bekannt, aber seine genaue rechtliche Situation und seine wirtschaftliche Bedeutungen sind allen noch nicht detailliert erkannt. Im industriellen Bereich sind Patente von großer Bedeutung. Speziell im IT Bereich gibt es heute zwei diskutable Themen: die Veröffentlichung der Kommunikationsprotokolle und die Patentierbarkeit von Software.

## Schlüsselworte
Patent, Softwarepatent, Kommunikationsprotokolle, Patentgesetz, Cross-licensing, Patent pool, Patent thicket, Patentstrategie

## 1. Einleitung
Der Patentschutz von technischen Erfindungen hat immer größere Bedeutung im industriellen Gebiet. Heute halten industrielle Unternehmen das Patent für die wichtige Wettbewerbskraft. Sie interessieren sich dafür, wie das aktuelle Patentsystem ist, welche Patentstrategie sie dafür auswählen sollen und wie sie das Patentmanagement erfolgreich durchführen können. Im IT-Bereich gibt’s zwei wichtige Themen: Veröffentlichung der Kommunikationsprotokolle und das Softwarepatent. In Europa und den USA bleibt bis heute noch eine wichtige Streitfrage, ob Software heutzutage patentierbar sein soll. Darüber sind die Softwareunternehmen verschiedener Meinungen. Anhand der rechtlichen und wirtschaftlichen Situation wird hier der Gründe für bzw. gegen Softwarepatente erläutert.

## 2. Das Patentwesen
Alle Länder haben eigene Patentgesetze, aber die Hauptziele vom Patentschutz sind gleich: die Motivation von technischer Entwicklung zu steigern, den wissenschaftliche und technische Fortschritt zu fördern. Mit dem Patentgesetz kann das ausschließliche Recht für den Erfinder gewähren lassen, damit der Erfinder bewilligt, die Erfindung veröffentlichen zu lassen. Die Veröffentlichung der Erfindungen gibt anderen eine Chance, die Weiterentwicklung durchzuführen, somit der technische Forschritt immer größer wird.

### 2.1 Ein historischer Überblick
Der Patentschutz war im Mittelalter erstens als ein von einer Autorität gewährtes Privileg. Die Gründe für diesen Gesetzschutz damals war aber nicht die Förderung des Erfindungswesens sondern die Förderung des Handelns. Insbesondere wurde im 14. Jahrhundert mit dem sogenannten „Einführungsprivilegien“ in Flandern und in Delft der „Außenhandel“ kontrolliert und dirigiert. Neben den Einführungsprivilegien gab es in der gleichen Zeit auch die ersten Erfindungsprivilegien, mit denen eine Ermächtigung zur Nutzung bzw. zur Herstellung der betroffenen Erfindung für den Erfindungsinhaber erteilt wurde. Die beiden Privilegien sind Begrifflich als die Vorgänger der heutigen Patent zu sehen. Aber inhaltlich können die beiden Privilegien und Patente nicht gleichgesetzt werden. [1] Mit der Definition des heutigen Patent ist der inhaltliche Unterschied dann einfach zu verstehen.

### 2.2 Definition
Das Patent bezieht sich auf das staatlich gewährte zeitlich begrenzte subjektive Ausschlußrecht, mit dem es verboten ist, die Dritten die patentierte Erfindung ohne Zustimmung vom Patentinhaber herzustellen, anzubieten, in Verkehr zu bringen oder zu den genannten Zwecken entweder einzuführen oder zu besitzen.
Nach § 1 Abs. 1 des Patentgesetzes (PatG) können Patente für Erfindungen auf allen Gebieten der Technik erteilt werden, sofern sie neu sind, auf einer erfinderischen Tätigkeit beruhen und gewerblich anwendbar sind. [2]

### 2.3 Rechtliche Beschränkungen
Wie die Definition beschreibt, wird das Patent vom Staat gegeben, d.h. das Patent gilt nur in dem Staat, der das Patente erteilt hat. Wenn der Patentinhaber in einem anderen Staat kein Patent für diese Erfindung erworben, kann die Erfindung nicht vom dem entsprechenden Staat geschützt. Diese Beschränkung wird „Territorialprinzip“ genannt. Außer der staatlichen Begrenzung wird das Patent auch mit einem bestimmten Zeitraum gebunden, laut § 16 PatG besitzt das Patent eine maximale Laufzeit von 20 Jahre ab Patentanmeldung. Nach dem Ablauf oder beim Fehlen der Zahlung der jährlichen Patentgebühr wird das Patent als das Allgemeingut vom Staat gesetzt. Weil das Nutzen vom Patent nicht ewig geblieben ist, wird diese zeitliche Beschränkung die Anreize zum Weiterentwicklung bzw. Bemühung um neue Forschungen des Patentinhaber versichern, damit die Förderung des technischen Fortschritt und optimale Wettbewerbssituation geschafft wird. [2]

Viele haben es immer missverstanden, dass der Patentinhaber mit der Patentanmeldung allerdings ein Recht von der Nutzung des Patents bekommt. Das Patent bezieht sich aber nicht auf ein Benutzungsrecht sondern ein Ausschlussrecht. Dieses Recht schließt die Anderen von der Verwertung des Patentsgegenstands ausgeschlossen, aber das bedeutet nicht, dass der Patentinhaber den Gegenstand selber verwerten kann. Laut PatG ist eine Voraussetzung zur Verwertung des Patents gegeben: solange die Verwertung den Recht von Anderen nicht entgegensteht, kann die patentierte Erfindung verwertet werden. [2]

Wie es in der Abbildung 1 bezeichnet: Die Fläche a, b und c bezeichnet die jeweiligen Bestandteilen von des Gegenstands der Patent 1, Patent 2 und Patent 3. Der Inhaber vom Patent 1 hat ein ausschliessliches Recht für den Bereich a. Wenn der Patent 2 angemeldet wird, hat sein Inhaber auch ein ausschließliches Recht für den Bereich b. Aber der Inhaber vom Patent 2 kann dieses Patent nicht verwerten, weil es eine Überlappung mit den Bestandteilen vom Patent 1 gibt. Er kann das Patent 2 nur verwerten, wenn er die Zustimmung vom Inhaber des Patents 1 hat. Ebenso kann der Inhaber des Patents 3 das Patent 3 auch nicht verwerten, ohne die Zustimmung vom Inhaber des Patents 2.

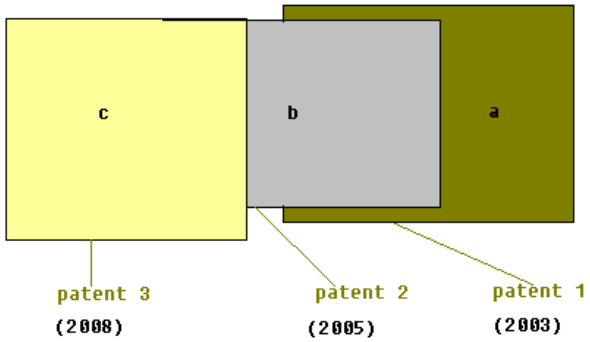



**Abbildung 1, Patentüberlappung**

# 3. Das aktuelle Patentsystem

Heutzutage ist es besonders wichtig für industrielle Unternehmen die richtige Patentstrategie auszuwählen und das richtige Patentmanagement durchzuführen. Das ist aber eine Aufgabe, die immer komplizierter wird. Wegen der rasenden Geschwindigkeit von technischen Entwicklungen, entsteht es zwischen den industriellen unternehmen ein starkes Konkurrenz zur Patentanmeldung. Die Anzahl der erteilten Patente haben sich immer schneller zugenommen, die häufig zum Problem mit Patentüberlappungen führt. So entsteht das sogenannte „Patent thicket“ (Patent-Dickichte). Die Kosten von Verwertungen der Patente werden immer höher. Dieses Problem fördert die Möglichkeiten der Kooperationen zwischen verschiedenen Unternehmen. Zwei häufige Lösungen zu diesem Problem ist: „Patent pool“ und „Cross licensing“. [3]

## 3.1 Cross licensing

Das „Patent Pool“ bezieht sich auf ein Abkommen für das Tauschen von einem oder mehreren Patenten zwischen zwei oder mehreren Patentinhabern oder die Übergabe der Patente an die dritten Parteien. Alternativ kann das „Patent-Pool“ auch als die Aggregation von Rechte des geistigen Eigentums definiert werden, welche der Gegenstand vom „Cross-Licensing“ ist. [4] Unter dem „Cross-Licensing“ kann man das Tauschen von besitzenden Patenten verstehen. Mit einem Abkommen einigen sich zwei Unternehmen die gegenseitige Nutzungserlaubnis von den im Abkommen betroffenen Patenten. Im IT-Bereich ist „Cross-Licensing“ die gewöhnliche Lösung zum „Patent thicket“ Problem. Insbsondere ist das „Cross-Licensing“ meistens eine Weise zu vermeiden oder lösen eines Rechtstreits. [5] D.h. wenn zwei Unternehmen eine selbe Technik entwickeln wollen, mit einem vorher vereinbarten Abkommen können mögliche auftretende Rechtsstreite vermieden werden. Anderseits kann ein auftretender Rechtstreit auch durch einen „Cross-licensing“-Vertrag gelöst werden.

Ein anderes Ziel vom „Cross-licensing“ ist das Erwerben vom fremden Know-How. Laut einer Umfrage-Studie von BDO im Jahr 2007 haben meiste deutsche Unternehmen die Meinung, dass Patente zur Monopolisierung von Märkten wichtig sind. Nur 44% sind interessiert am Erwerb der fremden Patente. [6]

## 3.2 Motivationen zu patentieren

Im Jahr 2002 hat das „Fraunhofer Institute for Systems and Innovation Research“ (Fraunhofer ISI) eine Recherche über das Thema „Motivationen zu patentieren“ durchgeführt. 500 deutsche Unternehmen nahmen an der Umfrage teil. Tabelle 1 zeigt uns zuerst, wie wichtig Patentstrategien als eine Schutzmöglichkeit für Unternehmen sind. „Patentstrategien im Ausland“ ist die zweitwichtigste Strategie. Und die drittwichtigste Strategie ist „Nationale Patentstrategien“. Bei diesem Umfrage-Ergebnis wird man auf die Wichtigkeit der Patentstrategien sehr aufmerksam. Ein besonderer Umstand ist es, dass wegen des Regelns vom Territorialprinzip die Patentstrategie im Ausland nach der Meinung von meisten Unternehmen wichtiger als nationale Patentstrategien ist. [7]

**Tabelle 1. Wichtigkeit der Schutzstrategie: im allgemein**

**(Der Anteil der Unternehmen, der die Strategie als wichtig bewertet) [7]**

| Strategie | Wichtigkeit |
|---|---|
| zeitliche Vorsprung | 0,88 |
| Patentstrategien Ausland | 0,79 |
| Patentstrategien Inland | 0,72 |
| Beziehungsmanagement | 0,66 |
| Markenschutz | 0,58 |
| Geheimhaltung | 0,58 |
| Gestaltung der Zuliefererverträge | 0,51 |
| Maßnahmen zur langfristigen Personalbindung | 0,44 |
| Gebrauchsmuster | 0,24 |
| Urheberrecht | 0,15 |
| Geschmacksmuster | 0,11 |

Um genau zu wissen, welche wichtigen Motive zu patentieren die meisten Unternehmen haben, führte das Fraunhofer ISI noch eine weitere Umfrage durch. Die Tabelle 2 stellt ein Resultat der Umfrage über die Wichtigkeit der jeweiligen Gründe zur Patentanmeldung dar. 84% der befragten Unternehmen halten den „Schutz vor Imitation“ als der wichtigste Grund von Patentanmeldungen. Weitere wichtige Gründe sind „Absicherung des Markts“, „defensive Blockierung“, „Sicherung nationaler märkte“, „Reputation“ und „offensive Blockade“. Die „defensiv Blockade“ bedeutet die Lassen alle 15 Motivationen zusammengefasst werden, ergibt es sich hauptsächlich fünf Motivationstypen bei der Patentanmeldungen: das protektive Motiv, das Blockademotiv, das Reputationsmotiv, das Tauschmotiv und das Aufmunterungsmotiv. Das protektive Motiv bezieht sich auf den Schutz der patentierten Erfindungen. Das Blockademotiv bezieht sich auf die „offensive Blockade“ und „defensive Blockade“. Die defensive Blockade bedeutet: Mit den Anmeldungen von Abhängigen Patente wird das eigene Basispatent von dem Unternehmen umgeringt. Ein Unternehmen kann eine solche Patentmauer um sein Basispatent errichten, damit die Konkurrenz ferngehaltet wird. Die offensive Blockade

heisst, dass ein Unternehmen viele abhängige Patente vom Basispatent der Konkurrenz anmelden. So wird das Basispatent der Konkurrenten von solchen abhängigen Patenten umgeringt. Im diesen Fall wird den Konkurrenten die Vermarktung der Basisprodukte erschwert. Das Tauschmotiv bedeutet: die Patente als eine Verhandlungsmasse zu nutzen. Mit der Anmeldung eines Patents bekommt ein Unternehmen nicht nur den Gesetzschutz, sondern auch die Tauschpotential. Z.B. Für das Vereinbaren der Cross-Licensing-Verträge ist einem Unternehmen die Anzahl besitzender Patente besonders wichtig. Das Reputationsmotiv bezieht sich auf das technische Image von einem Unternehmen. Dieser Motivationstyp zeigt, dass Patente als der interner Leistungsindikator von einem Unternehmen zu verstehen sind. Im industriellen Bereich haben Patente daher große Bedeutung für Unternehmensreputation. Das Aufmunterungsmotiv zeigt die Ermunterungswirkung der Patente für die Mitarbeiter eines Unternehmens.

**Tabelle 2. Wichtigkeit der Motive zum Patentieren (Der Anteil der Unternehmen, der die Motivation als wichtig bewertet hat) [7]**

| Motivation | Wichtigkeit |
|---|---|
| Schutz vor Imitation (traditionelles Motiv) | 0,84 |
| Sicherung europäischer market | 0,75 |
| defensive Blockade | 0,72 |
| Sicherung nationaler märkte | 0,71 |
| Reputation / technologisches Image | 0,69 |
| offensive Blockade | 0,69 |
| Sicherung außereuropäischer Märkte | 0,57 |
| Steigerung des Unternehmenswertes | 0,51 |
| Verbesserung der Position in Kooperationen | 0,39 |
| Mitarbeitermotivatioin | 0,32 |
| Tauschpotential (z.B. cross licensing) | 0,28 |
| Zugang zum Kapitalmarkt | 0,26 |
| Interner Leistungsindikator | 0,22 |
| Lizenzeinnahmen | 0,21 |
| Standardisierungsaktivitäten | 0,20 |

In der Studie von Fraunhofer ISI werden neben den eigenen Umfragensergebnissen auch empirische Forschungsergebnisse in früheren Jahren gezeigt. Die Tabelle 3 stellt das Ranking von wichtigsten Motivationen in den Jahren 1995 bis 2003. Was unsere Aufmerksamkeit erregt, ist der wichtigste Grund zu patentieren im Jahr 2003: „Zwang zu Patenten wegen Patentpraktiken anderer“. Die starke Konkurrenz im heutigen Patentsystem zwingt Unternehmen zur Patentanmeldungen. Dieser Umstand wird auch durch die zweitwichitigste Motivation „Verhandlungsmasse“ im Jahr 2003 indirekt gezeigt. Diese Umfrage zeigt zwei wichtige Tendenz bei der Patentanmeldung. Das Patent bedeutet heute nicht nur das rechtliche Schutz, sondern auch eine Verhandlungsmasse und eine Ertragsmöglichkeit der Unternehmen. Diese Motive entstehen als die Folge von der weit verbreiteten Patenthandels wie „Cross-licensing“, „Patent pool“ und Lizenzverträge. [7]

Die Hauptmotivationen haben sich schon geändert. Die wichtigsten Motivationen beinhaltet das traditionelle Motiv „Schutz vor Imitation“ nicht mehr. Unternehmen betrachten Patente heutzutage aus strategischer Sicht und wirtschaftlicher Sicht. Somit bringen Patente Unternehmen auch den abgeleiteten Nutzen und spielen eine wichtige Rolle bei Unternehmensstrategien. Diese änderung der Motivation sowie das abgeleitete Nutzen von Patenten haben auch einen großen Einfluss auf das ganze Patentsystem.

**Tabelle 3. Das Ranking von wichtigsten Motivationen zum Patentieren aus Empirischen Studien in früheren Jahren [7]**

| Motivation | 1995 | 1998 | 2002 | 2001 | 1999 | 2003 |
|---|---|---|---|---|---|---|
| Schutz vor Imitation | 1 | 1 | 1 | 1 | 1 | |
| defensive Blockierung | 3 | 2 | 3 | | 2 | 3 |
| offensive Blockierung | | | 2 | 2 | 3 | |
| Reputation / techn. image | | | 5 | | 6 | |
| Internationale Marktverweiterung | 5 | 5 | | | | 4 |
| interne Leistungsindikator | 6 | 6 | 7 | | 5 | |
| Tauschenspotential / Verhandlungsmasse | 2 | 2 | 4 | 3 | 4 | 2 |
| Lizenzeinnahmen | 4 | 4 | 6 | 4 | 7 | 5 |
| Eigene Erfindung zum Standard machen | | | | 5 | | |
| Kapitalmarkt | | | | | | 6 |
| Zwang zu Patenten wegen Patentpraktiken anderer | | | | | | 1 |

.

## 3.3 Die Tendenz im Patentsystem

Wegen der immer größeren Bedeutung von Patenten wollen die großen Unternehmen immer mehrere Patente sammeln, um einen Wettbewerbsvorteil zu streben. Die Tabelle 4 zeigt die 10 aktivsten Patentanmelder im Jahr 2008 in Deutschland .Siemens AG hat als einer der größten Patentsammler immer gestrebt, mehrere Patente anzumelden, weil nach seiner Analyse die Anzahl von Patenten einen Einfluss auf den Verkaufserlös hat. [8]

**Tabelle 4. Die 10 aktivsten Patentanmelder 2008 (Vom deutschen Patent- und Markenamt im Jahr 2008 veröffentlichte Patentdokumente ohne Berücksichtigung eventueller Konzernverbundenheiten)[9]**

| Anmelder | Sitz | Anmeldungen |
|---|---|---|
| Robert Bosch GmbH | D | 2645 |
| Siemens AG | D | 1741 |
| Daimler AG | D | 1279 |
| GM Global Technology operations Inc. | USA | 994 |
| Denso Corp. | Japan | 716 |
| Bayerische Motoren Werke AG | D | 632 |
| Continental Autonomie GmbH | D | 632 |
| Schaeffler KG | D | 605 |

| | | |
|---|---|---|
| ZF Friedrichshafen AG | D | 594 |
| Volkswagen AG | D | 592 |

In den USA sammelte Microsoft als ein Beispiel 2000 Patetente im Geschäftsjahr 2003. Der Vorsitzender Bill Gates sagte am Ende 2003 noch, dass Microsoft 3000 Patente im Jahr 2004 anmelden sollte. Er hat behauptete, dass die Anzahl der Patente eine wichtige Rolle beim Übertreffen der anderen Konkurrenten (z.B. IBM) spielt. [10] Aber diese Forderung nach der Patentzunahme bedeutet 60 Patente pro Woche. Das sieht unmöglich aus. Dafür erklärte aber der Microsoft-Chefanwalt Brad Smith, dass die anderen Unternehmen im IT-Bereich durchschnittlich zwei Patente pro ein Millionen US Dollar Forschungsaufwand erwerben können. Microsoft hat jedes Jahr einen Forschungsaufwand in Höhe von sechs bis sieben Milliarden US Dollar geplant. Das bedeutet, dass 3000 Patente pro Jahr möglich sein sollten. Diese Behauptung scheint richtig zu sein. Aber die Tatsache ist: Um die Anzahl zu erreichen, anmeldete Microsoft damals sogar die Patente wie „System and Method for Creating a note related to a phone call“ und „Adding and Removing White Space from a document“, die überhaupt nicht „innovativ“ sind. [11] In diesem Fall entstehen solche Trivialpatente beim unsorgfältigen Prüfen vom Patentamt wegen der Konkurrenz zur Patentanmeldung. Es sieht wie ein Paradoxon aus, dass Microsoft Patente als wichtig für Innovation hält, aber die uninnovativen Patente anmelden möchte. Das ist ein typisches Beispiel von dem Wettbewerb mit Patenten.

Andrew Grove, der ehemalige Vorstandsvorsitzender von Intel hat das aktuelle Patentsystem in USA kritisiert. Im Jahr 2009 sagte er bei der Rede für seine Ehrenpreis für sein Lebenswerk: Patente werden heutzutage immer weiterer von den betreffenden Produkte entfernt. 50 Jahre früher war der Erfinder selber der Hersteller. Aber heute ist es üblich zu sehen, dass die Patente nicht von den Inhabern sondern von Anderen verwertet werden. Die Patente werden selber die Produkte und Instrumente der Investitionen, die im gesonderten Markt gehandelt werden. Erfinder erwarten hohe Erlöse bei Investitionen. Das abgeleitete Nutzen der finanziellen Instrumente wurde auch umgehindert verwendet. Das verletzte das finanzielle System und führte zur Finanzkrise. Das heutige Patentsystem in USA ist auf dem gleichen Weg. [12]

## 3.4 Patente auf Kommunikationsprotokolle

Bei Kommunikationsprotokollen ist es üblich, dass die Programme von verschiedenen Unternehmen zusammenarbeiten. Daher ist es besonders wichtig, die verschiedenen Unternehmen Kommunikationsprotokolle gemeinsam zu nutzen. Eine aktuelle Tendenz ist: die Kommunikationsprotokolle zu veröffentlichen. Darüber haben die Unternehmen im IT-Bereich unterschiedliche Meinungen. Ein Unterstützer dieser „open patent“ Strategie ist Cisco. Im Jahr 2008 haben Cisco Systems inc., Alcatel-Lucent, Clearwire Corp., Intel Corp., Samsung Electronics und Sprint die Open Patent Allianz für WIMAX (Worldwide Interoperability for Microwave Access) gegründet. Diese Allianz will einen patent pool an WIMAX-Patenten aufbauen. Der Patent-pool sammelt die grundlegenden Patente, die beim Implementieren des WIMAX-Standards notwendig sind und bietet anderen Unternehmen den Zugriff an. Damit können die Lizenzkosten einzelner WIMAX-Technologien möglichst gering gehalten werden und die Verbreitung und Weiterentwicklung der WIMAX-Technologien vereinfachen [13].

Eine andere wichtige „Open patent“-Aktion von Cisco ist die Nutzungserlaubnis für das Dokument „export of structured data in ipfix“. Unternehmen die selber keine Forderungen gegen Cisco stellen, können gewisse patentierte Technologien kostenlos nutzen, wenn diese Technologien im von IETF adoptierten Standard beinhaltet und die entsprechenden Patentrechte für das Praktizieren des Standards notwendig sind [14].

Mit diesen zwei „open patent“ Aktionen zeigt Cisco uns, dass die Open Patent Strategie sehr bedeutungsvoll für die gemeinsame Entwicklung der IT-Technik ist. Neben der Open Patent Strategie hat Cisco noch die Open Source Strategie durchgesetzt. Im Jahr 2008 hat Cisco das Messaging-Protokoll „Etch“ als Open-Source veröffentlicht. Cisco will SOAP mit diesem Messaging-Protokoll ersetzen, weil die weiteren Funkitionen vom Etch den Einsatz des Protokolls über mehrere Plattformen vereinfachen können. [15] Die Veröffentlichung der Protokolle ermöglicht die problemlose Zusammenarbeit der Programme von verschiedenen Unternehmen. Und die Open- Source-Eigenschaften vereinfachen noch die Vermarktung und steigern die Interoperabilität zwischen allen IT-Unternehmen.

Im Vergleich zu Cisco nutzt Microsoft die Patente immer sehr offensiv. Daher hatte die EU-Kommission im März 2004 gegen Microsoft eine Kartellbuße von fast 500 Millionen Euro verhängt. Die EU hat außerdem dem Konzern auferleget, die Schnittstellen zum Betriebssystem Windows offenzulegen und diese an Wettbewerber zu lizenzieren sowie eine Windows-Version ohne Media Player anzubieten. Da Microsoft bis 2007 die Auflagen noch nicht erfüllt hatte, drohte EU zuletzt im März 2007 mit einem Zwangsgeld [16].

Wegen der vielmaligen Anklagen von EU veröffentlichte Microsoft endlich im Februar 2008 technische Dokumentation zu den Kommunikationsprotokollen für „Office 2007“. Im April 2008 wurden noch weitere Protokolle und Schnittstellen veröffentlicht. Inklusiv sind die Protokolle und Schnittstellen des Office SharePoint Server 2007 und Dokumentationen der Protokolle, mit denen der Exchange Server 2007 Kontakt zu Microsoft Outlook aufnimmt und der Protokolle mit denen die Office-Clients mit anderen Microsoft-Server-Produkten zusammenarbeiten. Insgesamt wurden bis April 2008 schon über 14000 Seiten veröffentlicht. Aber bei betroffenen Patenten muss die Gebühren bezahlt werden [17]. Obwohl Microsoft damals zahlreiche technische Informationen veröffentlichte, ist seine Absicht bisher noch unklar. Weil die technischen Informationen nur aus Zwang veröffentlicht wurden, bleibt die Frage noch offen, ob Microsoft seine Strategie schon wechselt.

# 4. Softwarepatente

Eine der vom Patentschutz ausgeschlossenen Erfindungen ist der Computerprogramm bzw. die Software. Es gibt seit Jahren eine riesige Diskussion in EU und USA darum, ob die Software patentierbar sein soll. Die meisten großen Software-Unternehmen wollen Software patentierbar sein. Aber die meisten kleinen und mittleren Unternehmen sind dagegen.

## 4.1 Rechtliche Situation

Software ist in Deutschland urheberrechtlich geschützt, aber nicht patentierbar. [18] Die Tabelle 5 zeigt die rechtliche Abgrenzung zwischen den zulässigen und unzulässige Anspruchsfassung im IT Gebiet, welche die Klarheit in Abgrenzung der Anspruchsformulierungen bringt [19]. Wie die Tabelle 5 zeigt, dass die Software allein nicht patentierbar ist, aber wenn Unternehmen Software zusammen mit Hardware als technische Einheit patentieren lassen, ist diese technische Einheit patentierbar. In diesem Fall können Unternehmen die Regelung der Unpatentierbarkeit der Software umgehen. Das führt zur großen Anzahl der softwarebezogenen Patente in Europa. So ist die gegenwärtige Rechtslage unbefriedigend, da es an Klarheit

und Rechtssicherheit auf dem Gebiet der Patentierung der softwarebezogenen Erfindungen mangelt. [20] In Deutschland bzw. in Europa wird die Abgrenzung zur Softwarepatenten immer unklarer. Die Anerkennung der Softwarepatente wird von Patentämtern verstärkt. Bis 2002 war die Anzahl der Softwarebezogene Patente schon 20000. D.h. ca. 2% der gesamten EPA-Patente waren softwarebezogen. [21] Und bis 2009 wurden die Anzahl der softwarebezogenen Patenten schon auf 50000 gestiegen. [22] Trotz der verstärkten Anerkennung der Softwarepatente von Patentämtern ist die Patentierbarkeit der Software in Europa ausgeschlossen. Im Juli 2005 wurde ein Konzept von der Richtlinie zur Vereinheitlichung der computerimplementierten Erfindungen von der EU-Kommission abgelehnt. Nur 14 Abgeordnete waren zugestimmt, während 648 Abgeordnete dagegen waren. [23]

**Tabelle 5. Teile der Patentierbarkeitstabelle [19] S.102**

| **Ausnahme** | **Beispiele für nicht zulässige Anspruchs-fassung** | **Beispiele für zulässige Anspruchs-fassung** | **Rechts-vorschrift aus dem EPÜ** |
|---|---|---|---|
| Software/ Computer | Programm für eine Daten-verarbeitungs-anlage mit den Schritten A+B.<br><br>A+B sind nicht technisch, bzw. gehen über die normale physikalische Interaktion zwischen Software und Hardware nicht hinaus. | Programm für eine Daten-verarbeitungs-anlage mit den Schritten A+B.<br><br>A+B sind technisch, bzw. gehen über die normale physikalische Interaktion zwischen Software und Hardware hinaus. | Artikel 52(2)c |
| | Analog: Ansprüche die auf computer-lesbare Medien oder Datenträger gerichtet sind. | Analog: Ansprüche die auf computer-lesbare Medien oder Datenträger gerichtet sind. | (Entscheid ung der Techn.Bes chwerde-kammer T1173/97) |
| | | Computer oder Computer-system beinhaltend Mittel zur Ausführung der Schritte A+B | Artikel 52(3) |

In den USA gibt es eine drastische Tendenz zur Anerkennung der Softwarepatente. Das Prinzip von der Patenterteilung in den USA ist eigentlich „neu, nützlich, nicht offensichtlich“. [21] Aber tatsächlich gibt es bisher schon eine unüberschaubare Anzahl der Trivialpatente als das Ergebnis von softwarebezogenen Patenten, welches nicht mehr dem „Geist“ vom Patentrecht entspricht [24]. Ein typischer Beispiel vom Trivialpatent ist die obengenannte „Erfindung“ von Microsoft: „Adding and Removing White Space from a document“.

## 4.2 Gründe für Softwarepatente

Die meisten großen Software Unternehmen wollen Software patentieren lassen. Ein Repräsentant davon ist Siemens AG, die schon eine Broschüre zur Erläuterung der Patentierbarkeit der Software erstellt. Die Broschüre hat zum ersten erläutert: der einzige Grund gegen Softwarepatente entsteht wegen des Missverständnisses der Eigenschaft von Software. Man glaubt, dass die Software keinen technischen Charakter besitzt, aber die Software könnte tatsächlich auch technisch sein. Außerdem wird es noch eine Kategorisierungsmethode dargestellt, mit der eine Erfindung gesichert werden kann, ob sie technisch ist. Die Siemens AG glaubt, dass der einzige Schutz vom Urheberrecht nicht ausreichend ist, da das Urheberrecht nicht alle denkbaren Programme für eine Problemlösung sondern nur ein einziges Programm schützt. [25] In einer anderen Zeitschrift von Siemens AG wird es auch betont, dass die Patente als der Motor des Fortschritts eine erhebliche Rolle bei der technischen Entwicklung spielen. Die Siemens AG behaupte sogar, dass ohne Patente jede technische Entwicklung ziellos sein wird. [8] Ähnlich wie Siemens AG meint Microsoft-Vorstandsvorsitzender Bill Gates auch, dass der Kreislauf der Innovation für das Wachstum des Unternehmens von großen Bedeutung ist. Und ein wichtiger Teil des Kreislaufs ist das Patent. [26] Die Unterstützer von Softwarepatenten behaupten hauptsächlich, dass die Forschung bzw. Erfindung sich lohnen müssen und Software nicht die Ausnahme sein sollte. Die Softwarepatente sind nicht nur nützlich für große Softwareunternehmen sondern auch für mittlere und kleine Software Unternehmen. Die mittlere und kleine Software Unternehmen können über eigene Patente die gleich berechtigte Kooperationspartner für große Softwareunternehmen werden und dadurch viele Vorteile besitzen. Eine Konsultation aus dem Jahr 2000 zeigt einen gewichtigen Grund für Softwarepatente: „Investitionen und Arbeitplätzen“. [23]

## 4.3 Gründe gegen Softwarepatente

Im Gegensatz zu den Unterstützern von Softwarepatenten haben immer mehrere Wissenschaftler, die kleinen und mittleren Unternehmen erläutert, dass die Softwarepatente den technischen Fortschritt stoppen werden.

Mit der Ablehnung des Konzepts von der Richtlinie zur Vereinheitlichung der computerimplementierten Erfindungen in EU wird es indirekt gezeigt, dass Softwarepatente ziemlich problematisch sind. Ein Grund gegen Softwarepatente ist: für Computerprogramme sind die Kosten und Zeitaufwand der erstmaligen Produktion immer sehr hoch, aber die Reproduktionskosten sind nahe zu null, und der Zeitaufwand sind nur einige Minuten. Das Vermeiden vom Plagiarismus ist nicht einfach. Aus diesem Grund glauben manche Unternehmen, dass der Schutz vom Urheberrecht schon ausreichend ist. [18]

Hinsichtlich der Situation in den USA ist ein Ergebnis von Softwarepatenten die unüberschaubare spezifische Trivialpatente, welche keine unsichtbare verfahren, sondern sichtbare Prozesse und Ideen sind. Immer einfachere Algorithmen und immer kleinere Softwaremodule werden patentiert. Das führt zur Gefahr

für Entwickler, unwisslich Patente der Anderen zu verletzen und dafür gerichtlich belangt zu werden. [24] Außerdem sind auch die Probleme beim Monopol von großer Bedeutung. Die mittlere und kleine Unternehmen haben weder „Patent Pool“ als Verhandlungsmasse zu benutzen noch große finanzielle Macht die Patentgebühren, Patentrechercheaufwand und Prozessrisiko zu leisten. Die Softwarepatente werden offensichtlich zum Missbrauchen der Monopolkräfte der großen Softwareunternehmen führen. Aus der effizienten Sicht sind die unzureichende Patentprüfungen, die Kosten für Patente und die Dauer der Patenterteilung auch fragwürdig. Das wird die Geschwindigkeit der Innovation drosseln. Obwohl die großen Unternehmen immer behauptet, dass es eine Möglichkeit zur Kooperation mit den kleinen und mittleren Unternehmen gibt, könnte es auch sein, dass die großen Unternehmen die Patente von kleinen und mittleren Unternehmen ignorieren. Die Softwarepatente können zur falschen Relation von Patentkosten und Entwicklungskosten führen. Die Investitions-/Materialkosten würden geringer, solange der einfache Marktzugang vorliegt. Dann wird die Patentanmeldung der Software keinen wirtschaftlichen Sinn machen. [21]

## 4.4 Schlussfolgerung von Softwarepatenten

Die aktuelle Situation vom Patentsystem in den USA zeigt, dass die Tendenz von Anerkennung der Softwarepatente mehr Probleme als Nutzen gebracht hat. Die unüberschaubaren Trivialpatente bringen Unternehmen die Gefahr. Das verhindert die Entwicklung der Open-Source-Software, so dass einige Programme nicht veröffentlicht werden können, welches gegen den Hauptziel des Patentgesetzes ist. Die Softwarepatente führen außerdem auch zu einer schlechten Konkurrenzsituation für kleine und mittlere Unternehmen. Daher werden die ungewünschten Monopolkräfte entstehen, die den technischen Fortschritt nicht beschleunigen sondern stoppen werden. Obwohl die großen Unternehmen immer betonen, dass ohne Softwarepatente die Entwicklung ziellos und die Motivation zur Entwicklung geringer werden, ist es offensichtlich, dass der technische Fortschritt heute immer größer und die technische Entwicklung immer schneller wird. Die Softwarepatente bringen klar nur ein großes Nutzen an großen Unternehmen aber nicht an die Öffentlichkeit. Daher kann es hier mit einer Schlussfolgerung gesichert werden, dass die Möglichkeit von Softwarepatente ausgeschlossen werden soll.

# 5. Zusammenfassung

Momentan ist eine Tendenz im Patentsystem deutlich: technische Erfindung zu veröffentlichen. Das beschleunigt die technische Entwicklung und steigert die Interoperabilität zwischen verschiedenen Unternehmen. Aber das aktuelle Patentsystem birgt auch einige Risiken. Die Bedeutung von Patente hat sich schon geändert. Das abgeleitete Nutzen von Patenten wird umgehindert verwendet. Das kann positiv für die Unternehmen sein, aber kann auch einen negativen Einfluss auf das ganze Patentsystem haben. Speziell im IT-Bereich wird die Tendenz, Kommunikationsprotokolle zu veröffentlichen, auch immer deutlicher. Die Softwareunternehmen haben diese Situation schon akzeptiert. Aber um weiteres Nutzen zu bekommen, haben die großen Softwareunternehmen immer die Forderung nach der gesetzlichen Erlaubnis der Softwarepatente gestellt. Trotzdem wegen des Nutzens der Öffentlichkeit bzw. Mehrheit ist es offensichtlich, dass die Verwirklichung der Softwarepatente zur Zeit nicht möglich ist und in der Zukunft auch nicht möglich werden soll.

# 6. Literatur


[1] C. Osterrieth, „Patentrecht“, 2007, 12-14

[2] R. Kraßer, „Patentrecht“, 2009

[3] C.Shapiro, ,„ Nevigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard Setting“, 2001

[4] J. Clark, „Patent Pools : A Solution to the Problem of Access in Biotechnology Patents“, 2000, http://www.uspto.gov/web/offices/pac/dapp/opla/patentpool.pdf, zugegriffen am 30.05.2009

[5] J. Fromm, „ Patent Pools and Cross-Licensing“, 2002, http://www.ftc.gov/opp/intellect/020417jefferyfromm.pdf, zugegriffen am 23.05.2009

[6] CIPOC, „Patentverwertung in der Praxis “, 2009, http://www.cipoc.com/090429_WUERZBURG-IHK_LOOP.pdf, zugegriffen am 10.06.2009

[7] K.Blind, „The Influence of Companies’ Patenting Motives on their Standardisation Strategies“, 2008, http://www.euras.org/uploads/2008presentations/blind_strategies.pdf, zugegriffen am 02.06.2009

[8] Siemens AG, „ Picture of the future“, Herbst 2001, http://w1.siemens.com/innovation/pool/de/Publikationen/Zeitschriften_pof/PoF_Herbst_2001/PoF_2-01_D_1203848.pdf, zugegriffen am 23.05.2009

[9] DPMA, 2009, http://presse.dpma.de/presseservice/datenzahlenfakten/statistiken/patente/index.html, zugegriffen am 20.05.2009

[10] Wolfgang Sommergut, 2004, http://sommergut.de/wp/archives/microsoft-will-ibm-bei-zahl-der-patente-ubertreffen/, zugegriffen am 23.05.2009

[11] The New York Times, „ Why Bill Gates wants 3,000 New Patents“, 2005, http://www.nytimes.com/2005/07/31/business/yourmoney/31digi.html?pagewanted=1&_r=1, zugegriffen am 23.05.2009

[12] TG Daily, „ Former Intel CEO criticizes US patent system“,2009, http://www.tgdaily.com/content/view/42294/113/ zugegriffen am 23.05.2009

[13] Open Patent Alliace, „Open Patent Alliance Formed to Advance WiMAX 4G Technology“, 2008, http://openpatentalliance.com/index.php?option=com_content&view=article&id=10:open-patent-alliance-formed-to-advance-wimax-4g-technology&catid=1:opa-newsroom&Itemid=5, zugegriffen am 10.06.2009

[14] IETF, „IPR Statement“, 2009, http://www.ietf.org/mail-archive/web/ipfix/current/msg04833.html, zugegriffen am 10.06.2009

[15] Network Computing, „Cisco ersetzt Soap mit eigenem Protokoll“ ,2008, http://www.networkcomputing.de/cisco-ersetzt-soap-mit-eigenem-protokoll/, zugegriffen am 18.06.2009

[16] Frankfurt allgemeinner.net, „Die EU kämpft gegen das Monopol“, 2007, http://www.faz.net/s/RubE2C6E0BCC2F04DD787CDC274993E94C1/Doc~E43DE5E4A369146E9B2BA5763E0D7EA1C~ATpl~Ecommon~Scontent.html, zugegriffen am 10.06.2009

[17] Network Computing, „Microsoft veröffentlicht Kommunikationsprotokolle“, 2008, http://www.networkcomputing.de/pdf/microsoft-veroeffentlicht-kommunikationsprotokolle/?no_cache=1 zugegriffen am 19.05.2009

[18] J.Ensthaler, „ Gewerblicher Rechtsschutz und Urheberrecht“, 2009, 117

[19] Weber/Hdemann/Cohausz, „ Patentstrategien“, 2006

[20] K.Blind, „ Wer braucht eigentlich Software-Patente? Ergebnisse einer empirischen Untersuchung“, 2001, http://www.friedewald-family.de/Publikationen/GI2001.pdf, zugegriffen am 01.07.2009

[21] A. Fink, „ Exkurs: Software-Patente“, 2002, http://wi.informatik.unibw-muenchen.de/C11/lectures-businessInformatics01/Document%20Library/wi3-ft02-vorl06-2auf1.pdf, zugegriffen am 30.05.2009

[22] Patentverein.de, „ Patent-Reformen“, 2007, http://www.patentverein.de/files/7patentreformen_q209.pdf, zugegriffen am 30.06.2009

[23] Politik-Digital.de, „ Sind Softwarepatente wirklich vom Tisch?“, 2005, http://politik-digital.de/edemocracy/netzrecht/softwarepatvomtisch050707.shtml, zugegriffen am 03.06.2009

[24] berliOS, „Open-Source-Software“, http://oss-broschuere.berlios.de/broschuere/broschuere-de.html, zugegriffen am 30.06.2009

[25] Siemens AG, Broschüre, http://www.dfki.uni-kl.de/~klein/Patente_Software.pdf, zugegriffen am 01.05.2009

[26] Silicon.de, „ Microsoft will mehr Patente anmelden“, 2004, http://www.silicon.de/cio/b2b/0,39038988,39171159,00/microsoft+will+mehr+patente+anmelden.htm, zugegriffen am 03.06.2009

# SNMP - Simple Network Management Protocol

Rene Brogatzki
Betreuer: Marc-Oliver Pahl
Seminar Innovative Internettechnologien und Mobilkommunikation SS2009
Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: brogatzk@in.tum.de

***Abstract*— Today's networks grow in size and complexity. This includes physical devices like switches, routers or hosts and different types of protocols that deploy a networking environment. Considering many thousands of these components rises the problem of maintaining network performance. This document gives an introduction and solutions to the task of network management.**

***Index Terms*— Network Management, Structure of Management Information, SMI, Management Information Base, MIB, Simple Network Managemet Protocol, SNMP**

## I. Introduction

Networks grow in size and complexity. In addition to this growth the number and variety of different components of modern networks (e.g. the Internet or corporate networks) increases as does the number of vendors. Every vendor developes its own network concepts and configuration tools. The increasing size of networks being administered demands more effort and money, to a point where a network might not be maintainable with affordable efforts concerning availability, security, manageability and quality of service. The Simple Network Management Protocol (SNMP) is intended to automate the task of network management.

This document is intended to give an introduction to problems, concepts and solutions of network management and an overview of the Simple Network Managment Protocol. The first section discusses the basics of network managment. The second section shows the development history of SNMP. Section three talks about the SNMP framework and takes closer look at its components. Section four gives information about practical apspects of network management with SNMP.

## II. Basics of Network Managment

The importance of network management will be examplified by three real world scenarios:

A network administrator is assigned with the responsibility of a network. When a device or service fails an employee or customer might call in order to inform the IT department. The administrator will look for the problem and its solution. In this case the administrator acts *reactively*. Service unavailability causes high costs depening on duration and severity. If tools or services are available to the network administrator to discover malfunctions before they happen, precautionary measures can be taken before devices/services fail. This is called *proactive* action.

In order to be able to act proactively technology that is capable of indicating problems (monitoring) is required. Another example: An administrator observes traffic flows between network segments and discovers that moving a server from one segment to another could reduce overall network traffic. In this scenario information is gathered with e.g. a network sniffer. Observing a network with a network sniffer consisting of thousands of devices/services is not feasible.

A last example: A network administrator wants to be informed about suspicious traffic to specific hosts or ports. This could originate from an introuder, an attack or a port scan. *Prevention* of network security breaches is generally preferable to damage recovery.

The International Organization of Standardization (ISO) defined a model that describes different areas of network management in a structured way shown below. (The description is not intended to be exhaustive)

### A. Performance Managment

The goal of performance management is measurement and analysis of link quality, network throughput and overall quality of service of network devices. Performance Management implements information gathering and storage. This is a central aspect of SNMP as will be shown later.

### B. Fault Management

The goal of fault management is recognition, logging and elimination of error situations. Fault management can be understood as immediate response to faulty network conditions. The basis for appropriate fault management is well planned performance management. Gathering and storing is a key part of fault management so the border between fault management and performance management is smooth. SNMP facilitates fault management.

### C. Configuration Managment

Configuration management enables network managers to monitor existing hard-/software and their configuration and to alter the configuration of devices on demand. This is a capability of SNMP.

### D. Security Management

The goal of security management is controlling access to resources according to previously defined rules or policies (e.g. Common Creteria). Parts of security management are key infrastructures for cryptographic services and firewalls. Observation of services and devices can be implemented by SNMP.

### E. Accounting Management

The goal of accounting management is to collect device and service access statistics for billing purposes. SNMP is rarely used for accounting management but monitoring information can be used for accounting purposes.

## III. History of the Simple Network Management Protocol

This section shows the evolution of SNMP from version 1 over mutliple instances of version 2 to version 3, the latter being the current version of the Simple Network Management Protocol

In the late 1980s the internetworking community realized the demand of a coherent network management framework implementing the network management model described in the previous section.

At that time no unified standard existed. Protocols in use were the High Level Entity Management System (RFC 1021), the Simple Gateway Monitoring Protocol (RFC 1028) and the Common Management Information Protocol (CMIP ITU-T X.700). The mentioned protocols were either complex or designed for special purposes.

The Internet Architecture Board (IAB) released RFC 1052 "IAB Recommendations for the Development of Internet Network Management Standards". In this document the IAB set the general requirements for an Internet Standard Management Framework and assigned the Internet Engeneering Task Force (IETF) to work on an Internet Standard Management Framework. The assigned workgroup was to create a draft within 90 days. An easy framework was demanded that could be implemented and adopted by everyone who needed to address network management tasks. This means that no special requirements should be needed in order to use the framework. The design should be based on CMIP of the ISO to keep the design process short.

The next three sections describe the design process from version 1 to the current version 3 and briefly introduce key concepts of the framework

### A. SNMPv1

Basic design decisions of the IETF for the Internet Standard Management Framework were based on the following four principles. The first principle is the separation of information and communication, which means that the protocol operations should be independent from the information transmitted avoiding complexity of the protocol itself (e.g the protocol operation should not change whether an integer or an IP address is submitted).

The second principle is to abstract and describe managed information in a consistent way not dependent on the information type.

The third principle is keeping the architecture as a whole modular to be able to developing or changing parts of the architecture without changing the whole framework.

The fourth goal was the demand of ease of implementation.In 1988 the definition of SNMPv1 was released in three RFCs.These RFCs document the basic components of the SNMPv1 framework. The first document describes the Structure of Management Information (SMI) [1] which is an abstract description language based on a subset of the Abstract Syntax Notation One (ASN.1). ASN.1 is a description language used for defining data and transmission of data. The SMI is used to describe the format of information transmitted and stored by SNMP. To represent, store and transmit information the second document defines the Management Information Base (MIB) [2]. The MIB in SNMPv1 consists of objects. An object is the smallest entity of information defined by the SMI. The third document describes actual protocol operations [3]. These three documents are the basic components of the architecture. They are independent from each other, as a consequence of the goal of a modular design. The three core parts will be further discussed in Section IV

### B. SNMPv2

SNMPv1 was accepted by network administrators, as it solved the problems with network management. The functionality and mechanisms of SNMPv1 were revised and security weaknesses were discovered. Plain text transmission of passwords (Community Strings), a lack of authorisation, a lack of integrity checks and missing replay protection were points criticised by the community. These security flaws did not arise from a lack of knowledge rather than assumptions made during the design: The working group of the IETF assumed that information read and transmitted would not reveal information relevant to a possible attacker. Writable information was not considered to control fundamental devices/services of the network. The second assumption was that the aforementioned Community Strings would serve the need for security as they take the role of passwords for the managed network.

In consequence three aditional RFCs(1351,1352,1353) were released. These documents describe methods for authentication, integrity, privacy, authorisation. These extensions are known as SNMPsec. The focus on security made it more secure than its predecessor but did not achieve acceptance and was superseded by SNMPv2 [4]. SNMPv2 updated the Management Information Base(MIB-2) [5], which introduced the capability of grouping single MIB objects to represent devices as a whole. This enables network managers to define MIB entries for a kind of device/service and to reuse them for similar devices/services. The Structure of Management Information was revised to version 2 (SMIv2) to address the definition of MIB groups.

The development community was not able to achieve consensus concerning the security model to be used with version

2. Consequently it divided into different groups. One group implement a security model based on Community Strings. A Comunity String is a password which grants either read only or read-write acces permisson. Community String names are transmitted in clear text. Any attacker auditing the network traffic can read the name from passing traffic enabling him to view or alter SNMP information. Another group implemented the User-based Security Model (USM). This model makes use of a username with two associated cryptographic keys and protocols (e.g. DES,MD5). Both an agent and a NMS need to know the same username and its associated security context in order to exchange information. This lead to the following different instances of SNMPv2: SNMPv2, SNMPv2c, SNMPv2u and SNMPv2*. The different versions of SNMPv2 did not get adopted. [6] The disunity of the developers is considered to be the cause of this rejection. [6]

### C. SNMPv3

The SNMP work group learned from the mistakes made when revising SNMPv1 and adapted to the demand for different security mechanisms. The SMIv2 and MIB2 got adopted. Further improvements were applied to the actual communication protocol. The security model may be USM or any older security model either with or without implementing the View-based Access Control Model (VACAM). VACAM handles access depending on the following factors: The username and group for general access control, where permission is bound to the group, the security protocol used for communication and permissions of the MIB. [7]. The second version of the communication protocol was adopted from SNMPv2 as was the MIB-2 and SMIv2. [6] Additional tools were added to assist the administration of SNMP. The standardisation process was finished in 2002 with RFC 3410-3418.

### D. Remarks on versioning of SNMP

Usually a newer version of a RFC obsoletes older versions. This was not the case with further improvements of the SNMP framework, e.g. SNMPsec did not obsolete SNMPv1. This was addressed later, but did not change the fact that SNMPv1 is the most widely used version of the SNMP framework. This holds true for today. Though SNMPv3 is considered to be the technically most mature version.

## IV. Simple Network Management Protocol Framework

The following sections describe the infrastructure of network management with SNMP. This describes the layout of a managed network and the different roles the single participants assume. The subsequent sections show the specific elements of the SNMP framework in detail. These are the SMI, the MIB and the protocol operations. The closing subsection will introduce basic security features SNMP is capable of.

Network management in context of SNMP is the ability to gather, store and alter information when needed. To acomplish this task the three aforementioned modules of SNMP work together, while using the infrastructure of the managed network itself, raising two problems: Dispatching too many network management tasks might affect network throughput and dispatching too less might lead to loss of important information. This decision depends on the network administrator [8]

The following four sections describe the core parts of SNMP beginning with an overview of different roles of participants of a SNMP network.

### A. Infrastructure of network managment

A managed network consists of three basic parts. A host taking the role of the management station, a number of managed devices/services, called managed nodes, and the actual protocol operations enabling communication between stations (discussed later in this section). Fig. 1. gives an overview of this concept:

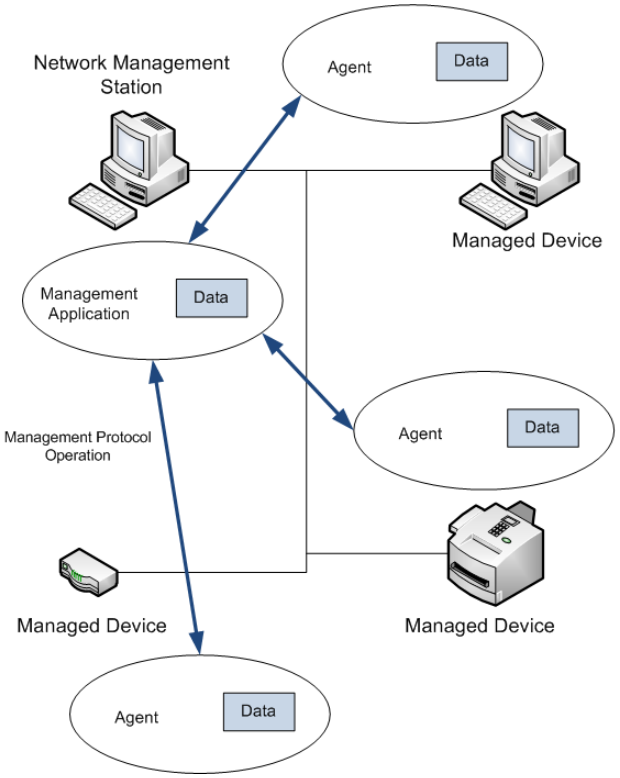



Fig. 1. Infrastructure of Network Management

The Network Management Station (NMS) consists of two software components: The management application - the interface between the managed network and the administration staff - and the database holding the managed information. This station collects, analyzes, processes and displays information from managed nodes as needed. The NMS reacts to the information collected either automatically or requested by the administrator in order to control the managed nodes of the network. The database of the NMS is a collection of MIB objects of managed nodes. This collection does not need to be exhaustive. If information is needed the NMS polls managed

nodes for it. All information centers at the NMS so the NMS is the place where network management takes place. [8]

Managed nodes are either physical devices or software services e.g. a webserver that is managed by the NMS. In order to facilitate network management functionality managed nodes include a sofware service called the agent and a set of MIB objects that describe the manageable information. The agent is the interface between the NMS and the remote physical device or software service. It receives incoming messages, processes them and dispatches answer messages, e.g. the NMS wants to know the uptime of the host therefore sending a message to the node to get the required data. The node receives the message, looks in its database for the appropriate value and generats a response message with the desired data and sending it to the NMS. [9]

Each node of a managed network holds managed information in the MIB in the form of single objects. The format of each object is defined by the SMI. The next two subsections will therefore introduce both concepts of SNMP in more detail.

### *B. SMIv2 and MIB2*

The Structure of Management Information (SMI) is a language that is used to describe the format of managed information. It ensures that syntax and semantics of managed data are well defined and unambiguous. The SMI itself does not define management data of a managed system, it rather defines the format. As with object oriented programming the concept of a class relates to the abstract definition of a data type which is not usable until instantiated. The SMI is the syntax to define the format of managed information, whereas the instantiation of a class resulting in an usable object corresponds to the previously defined information as a MIB Object. To be able to describe a broad variety of manageable devices the SMI contains the necessary syntactical elements and data types (e.g. IpAddress, Counter, NetworkAddress, Integer, Octet String). The syntax below shows the exact definition of a MIB object that counts the successfully delivered datagrams to IP user protocols. Each MIB Object follows this scheme.

```
ipSystemStatsInDelivers OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
"The total number of datagrams successfully
    delivered to IP user-protocols"
::= { ipSystemStatsEntry 18 }
```

The individual syntactical elements of the example above will be explained briefly. Object identifier (OBJECT-TYPE) defines the name of the object. SYNTAX defines the data type of this object. In this case it is a 32-bit counter. MAX-ACCESS describes access control. It is either read-only or read-write. The STATUS element is used to indicate whether the object is up to date, obsoleted, mandatory or optional. Individual objects are revised over time and eventually obsolete so that they should not be implemented. The DESCRIPTION element contains human readable description text. The sample above is representative for the design of all basic MIB objects.

MIB objects are arranged in a hierarchical tree structure where a single object can be thought of as a variable of managed information. Fig. 2 shows a part of a tree to visualize its layout. Single objects are referenced via the number or the name separated by a dot. The IP module is addressed by the Objec identifier (OID) 1(ISO).3(org).6(dod).1(internet).2(mgmt).1(MIB-2).4(IP). The OID of the example object above is 1.3.6.1.2.1.4.18. The addressing is relevant when looking in more detail at the protocol operations, since a read-message would address managed information via the combination of the OID and the corresponding value of the object.

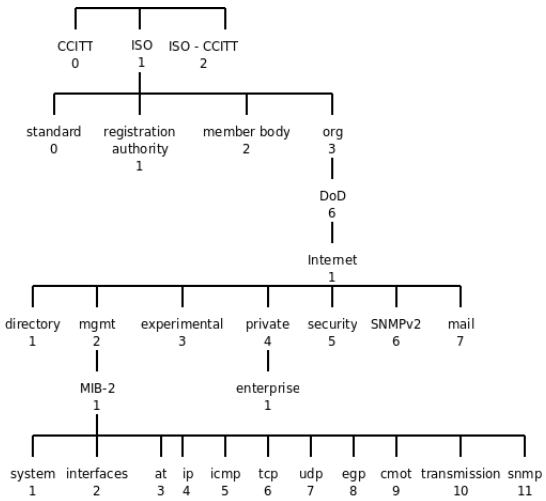



Fig. 2. Example view of the MIB Tree

With the next section the protocol operations shall be discussed to illustrate how network management with SNMP works in regards to communication between the NMS and managed nodes.

### *C. SNMP Protocol Operation*

The role of SNMP is to grant access to management information requestable over the network. This includes read and write operations and an alert notification. This allows an administrator to look for or alter specific information. In order to automate the functionality of network management, SNMP uses two policies. The first policy constitutes polling for reading/writing access. Only NMSes can poll agents for values of objects and instruct them to alter one or more objects. The second policy enables an agent to inform the NMS autonomously of special events (alarms, interrupts). With the NMS polling an agent for read/write operations the communication model can be considered a server client model, where the agents act as servers since they respond to messages sent by a distant peer - the NMS.

The SNMP protocol operations reside on Layer 7 of the ISO/OSI referrence model. As transport service UDP is used on port 161 for read, write and response messages. Port 162 is used for trap messages. Only NMSes are supposed to listen

for traps. On the network layer the Internet Protocol is used. The payload of SNMP will be referred to as Protocol Data Units (PDU). [8]

When a NMS requests information it sends a GetRequest-PDU. The PDU consists of the address, an identifier of the message, which is an integer used to relate replys of managed nodes to requests sent previously by a NMS. This represents the header of the message. The payload of a Request-PDU contains the the OID of the requested object. The agent looks for the OID in its MIB on arrival of a message takes the value bound to the OID and sends a Response-PDU to the NMS. The NMS is now able to update the local MIB at the OID with the value sent. When a NMS needs more than one OID it can send an GetBulkRequest-PDU. The structure of the GetBulk-PDU is the same as with the GetRequest-PDU despite the fact that it contains more than one OID. The agent will respond the same way but the response contains more bindings of OID and value. If the agent could not locate a requested OID it is responding with a Response-PDU containing an error code describing the error condition.

For writing purposes the NMS sends SetRequest-PDUs. The structure is the same as with GetRequest-PDUs but it contains a value associated with an OID. The agent has to sent an response with either no error code indicating a successful change of the value or an error code describing the reason.

As mentioned before SNMP was designed to be flexible and usable on any kind of network therfore it is possible to use different transportation protocols, e.g. AppleTalk,TCP or IPX. [10]

### *D. Security*

Considering a mechanism to monitor and control a network the mechanisms have to be protected agains possible attackers. Therefore SNMPv3 tries to achieve a set of security goals: Privacy is addressed with encryption of the SNMP-PDUs. The cyphersuite used is the Data Encryption Standard (DES) or Advanced Encryption Standard (AES)in Cipher-Block-Chaining-Mode (CBC). The administrator has to distribute the keys to all participating nodes.

Authentication is addressed with the utilization of a cryptographic hash function (e.g. MD5) and a secret but shared key. This mechnism is know as Hashed Message Authentication Code (HMAC). This mechanism concatenates the message with the secret key, which does not need to be the one used with DES, and then hashed. The hash value is concatenated to the message and sent. The receiver knows a key and is able to recalculate the hash. Do the results match with the transmitted HMAC the sender is authenticated. As a side effect this effect ensures the integrity a message, which means it can be decided if the message was altered during transfer.

Another security issue addressed by SNMP is the protection against reinjected old messages. This is called replay protection and might lead to inconsitent MIBs, which could lead to erroneous actions taken by the NMS. Therefor a value representing the uptime of the system is calculated and used as a timestamp. [8]

## V. Conclusion

The Simple Network Management Protocol Framework is a tool for managing networks consisting of different devices and protocols. The modularity makes it extensible and flexible. The mistakes made with the design of version two are still visible as the insecure version 1 is still the most used one. An explanation might be the multiple, partially not standardised, revisions of version two. In comparision to other management tools and frameworks SNMP is the most widely deployed one of all available solutions, nevertheless newer technologies exist, e.g. Netconf, which focuses on managing configuration. A varity of tools implementing SNMP exist open source as commercial ones. Two examples of open source SNMP implementations are Net-SNMP and Nagios. Both implement the entire IETF framework specification.

## References


[1] M. Rose and K. McCloghrie, "Structure and identification of management information for TCP/IP-based internets," RFC 1155 (Standard), Internet Engineering Task Force, May 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1155.txt

[2] K. McCloghrie and M. Rose, "Management Information Base for network management of TCP/IP-based internets," RFC 1156 (Historic), Internet Engineering Task Force, May 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1156.txt

[3] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple Network Management Protocol (SNMP)," RFC 1157 (Historic), Internet Engineering Task Force, May 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1157.txt

[4] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Introduction to version 2 of the Internet-standard Network Management Framework," RFC 1441 (Historic), Internet Engineering Task Force, Apr. 1993. [Online]. Available: http://www.ietf.org/rfc/rfc1441.txt

[5] K. McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets:MIB-II," RFC 1213 (Standard), Internet Engineering Task Force, Mar. 1991, updated by RFCs 2011, 2012, 2013. [Online]. Available: http://www.ietf.org/rfc/rfc1213.txt

[6] C. M. Kozierok, "The TCP/IP Guide," Book, Oct. 2005.

[7] B. Wijnen, R. Presuhn, and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)," RFC 3415 (Standard), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3415.txt

[8] J. F. Kurose and K. W. Ross, "Computernetwerke," Book, 2008.

[9] J. Case, D. Harrington, R. Presuhn, and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)," RFC 3412 (Standard), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3412.txt

[10] R.Presuhn, "Transport Mappings for the Simple Network Management Protocol (SNMP)," RFC 3417 (Standard), Internet Engineering Task Force, Dec. 2002, updated by RFC 4789. [Online]. Available: http://www.ietf.org/rfc/rfc3417.txt

[11] J. Case, R. Mundy, D. Partain, and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework," RFC 3410 (Informational), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3410.txt

[12] D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411 (Standard), Internet Engineering Task Force, Dec. 2002, updated by RFC 5343. [Online]. Available: http://www.ietf.org/rfc/rfc3411.txt

[13] D. Levi, P. Meyer, and B. Stewart, "Simple Network Management Protocol (SNMP) Applications," RFC 3413 (Standard), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3413.txt

[14] U. Blumenthal and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)," RFC 3414 (Standard), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3414.txt

[15] R. Presuhn, “Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP),” RFC 3416 (Standard), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3416.txt
[16] R.Presuhne, “Management Information Base (MIB) for the Simple Network Management Protocol (SNMP),” RFC 3418 (Standard), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3418.txt
[17] R. Frye, D. Levi, S. Routhier, and B. Wijnen, “Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework,” RFC 3584 (Best Current Practice), Internet Engineering Task Force, Aug. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3584.txt

# Webservices for embedded systems

Thomas Riedmaier
Betreuer: Andreas Müller
Seminar Innovative Internettechnologien und Mobilkommunikation SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: riedmaie@in.tum.de

***Kurzfassung*— Das „Internet der Dinge“ benötigt ein sauber definiertes Design um hersteller- und plattformunabhängige Interoperatibilität zu ermöglichen. Ein sehr vielversprechendes, und wegen der sowohl freien als auch breiten Verfügbarkeit seiner Basistechnologien in der Praxis oft verwendetes Konstrukt, wurde mit der Webservice Technologie entwickelt. Da diese jedoch klassischerweise auf Servern und anderen „thick clients“ zum Einsatz kommt, stellt sich die Frage, ob - und mit welchen Anpassungen - sie auch von sogenannten „embedded systems“ genutzt werden kann. In dieser Arbeit wird sowohl dieser Fragestellung nachgegangen, als auch eine Diskussion zu den Alternativen der Webservice Technologie geführt.**

***Schlüsselworte*— Webservices, embedded systems, DPWS, SOA, WS-***

## I. Einleitung

Das Türschloss redet mit der Kamera, der Kühlschrank fragt bei den örtlichen Supermärkten nach, wer frische Erdbeeren im Angebot hat und der Küchenroboter macht sich gerade schlau, wie man einen Tisch für ein „Dinner for two“ deckt. Zukunftsmusik? Fakt ist, dass das klassische Web-Nutzungszenario eines menschlichen Nutzers, der mit Hilfe eines Webbrowsers auf Angebote zugreift, immer mehr an Bedeutung verlieren wird.

Der Trend geht zum „Internet der Dinge“. In diesem neuen Internet tritt der Mensch nicht mehr selbst als Dienstnutzer auf. Vielmehr kommunizieren eine Vielzahl unterschiedlichster Geräte selbstständig miteinander, wobei sie gegenseitig ihre Dienste in Anspruch nehmen, um eine für den Menschen nützliche Funktionalität zu erbringen.

An dieser Stelle wäre es natürlich wünschenswert, wenn es Standards gäbe, so dass auch Entwicklungen unterschiedlicher Hersteller miteinander kommunizieren können. Mit dem Webservice-Framework wurde von *der* Standardisierungsorganisation des Webs, dem W3C, eine Reihe genau solcher Standards entwickelt. Da das WS-Framework auf den - ebenfalls vom W3C spezifizierten - weit verbreiteten und frei verfügbaren Webtechnologien wie HTTP oder XML basiert, erfreut es sich großer Beliebtheit in der Entwicklergemeinde. So verwundert es nicht, dass bereits heute eine große Anzahl von Webservicen - zum Teil auch öffentlich - verfügbar sind [1]. Primär wurde das Framework jedoch für Unternehmensnetzwerke entwickelt, weshalb die Bedürfnisse von Systemen mit beschränkten Ressourcen wenig Beachtung fanden. Genau hier könnte sich ein Problem auftun: ein Standard der für einen Webserver Sinn macht, muss noch lange nicht auf einem eingebetteten System („embedded system“) wie einem Kühlschrank realisiert werden können. Wie es dennoch möglich ist, Webservices nicht nur für PCs und Server zu entwickeln, sondern auch für eingebettete Systeme wird in dieser Ausarbeitung beschrieben.

Zu diesem Zweck wird zunächst in Kapitel II eine kurze Einführung in diejenigen Konzepte gegeben, welche in Kapitel III nötig sind um das Prinzip der Webservice Architektur nachvollziehen zu können. Aufbauend darauf wird in IV mit dem DPWS die ausgereifteste Anpassung der Architektur für eingebettete Systeme vorgestellt. Abschließend folgen in Kapitel V Alternativen zu diesem Ansatz sowie die Diskussion, ob nicht andere Technologien viel besser für das Anwendungsgebiet der eingebetteten Systeme geeignet wären.

## II. Begriffsklärung

Das Konzept „Webservices for embedded systems“ lässt sich viel besser nachvollziehen, wenn man sich zunächst noch einmal klar macht, was hinter den Begriffen „Web“(Abschnitt II-A), „Service“(Abschnitt II-B) und „embedded system“(Abschnitt II-C) steckt.

### A. „Web“

Das *World Wide Web(WWW)* - oder kurz Web - stellt innerhalb des Internets ein globales Informationssystem dar, welches es seinen Benutzern ermöglicht, auf sich in ihm befindliche Ressourcen zuzugreifen. Mit dem Begriff Ressource sind bei weitem nicht nur HTML-Dokumente gemeint, sondern im Prinzip alles, was sich durch einen *Uniform Resource Identifier(URI)* identifizieren lässt wie zum Beispiel eben auch ein eingebettetes System [2].

Da es nicht möglich ist, auf das Abstraktum einer Ressource zuzugreifen, benötigt jede Ressource eine Repräsentation. Wie diese auszusehen hat, wurde beim Entwurf des WWW nicht festgelegt. Auf diese Weise konnten sich neben dem ursprünglichen HTML-Format nach und nach

noch eine Reihe weiterer Formate etablieren. Exemplarisch seien hier RDF/XML, XHTML und CSS genannt [2].

Während HTML sich vorrangig zur Mensch-Maschine-Kommunikation eignet, stellen die Formate der XML-Familie eine Möglichkeit zur strukturierten Maschine-Maschine-Kommunikation dar [3].

Womit auch schon die letzte zentrale Komponente des WWW thematisiert wäre: Die Übertragungsfunktionalität für die Ressourcenrepräsentationen. Diese Funktionalität basiert auf dem Verschicken von Nachrichten auf Anwendungsebene. Das Protokoll, nach welchem dies stattzufinden hat, ist jedoch nicht festgelegt. In der Praxis werden - je nach Anwendungsgebiet - überwiegend die Protokolle HTTP, FTP oder SMTP eingesetzt [2], [4].

### B. „Service“

Als Dienst (engl. „Service“) wird in dieser Arbeit eine Sammlung von Operationen bezeichnet, die von einem Rechensystem auf Anfrage hin durchgeführt werden können.

Der Ansatz, die Funktionalität eines Softwaresystems nicht mehr in Form von Bibliotheken, Klassen, oder Ähnlichem zu bündeln, sondern sie vielmehr als z.B. über das Web erreichbaren Dienst zur Verfügung zu stellen (Abb. 1), wird als *Service Oriented Architecture (SOA)* bezeichnet.

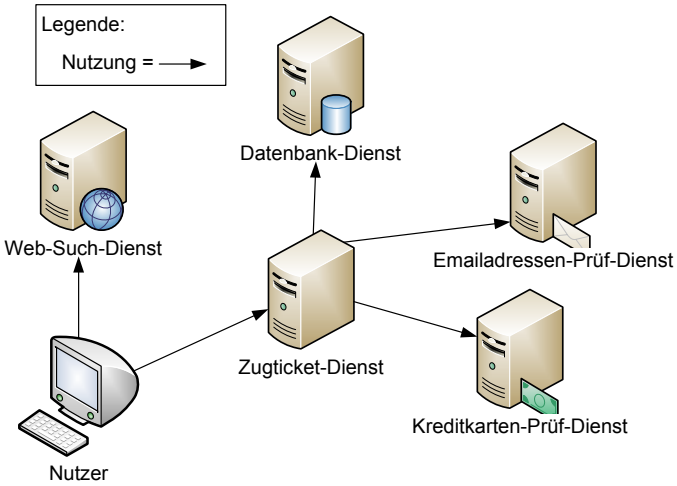



Abbildung 1. Visualisierung einer Beispiel-SOA

Diese Technologie versteht sich selbst als *das* Mittel der Wahl um viele Anforderungen, die an die moderne Softwareentwicklung gestellt werden, zu erfüllen. Zu diesen Anforderungen gehören [5]:

- Maximale Wiederverwendbarkeit
- Möglichst gute Wartbarkeit
- Einfachheit und Modularität der Systeme

### C. „Embedded system“

Bei einem eingebetteten System (engl. „embedded system“) handelt es sich um ein Computer-System, das Teil eines größeren Gesamtsystems, wie z.B. eines Kühlschranks ist. Charakterisierend für diese Systeme ist, dass sie aus Kostengründen meist nur mit sehr leistungsschwacher Hardware ausgestattet werden [6].

## III. Die Webservice Architektur

Aktuellen Schätzungen zufolge gibt es derzeit weltweit knapp 1,6 Milliarden Internetnutzer, Tendenz stark steigend [7]. Außer des Stromnetzes dürfte es damit kein Netzwerk geben, welches mehr Menschen zur Verfügung steht, was dem Internet den Titel des am weitest verbreiteten Datennetzwerkes verleiht. Es bietet sich also an, wenn man sich zum Ziel gesetzt hat eine möglichst universell einsetzbare SOA zu entwickeln, diese für das Internet auszulegen und sich der in ihm verfügbaren Systeme wie dem WWW zu bedienen. Die Umsetzung dieser Idee wurde ausgehend von den Technologien, die für diese spezielle Realisierung einer SOA verwendet werden, auf *Webservice Architektur (WSA)* getauft.

Die WSA verwendet nur eine spezielle Art von Diensten: die *Webservices(WS)*. Als WS werden hierbei allgemein Dienste bezeichnet, die auf dem WS-Framework aufbauen. Das WS-Framework wiederum ist der Versuch, eine Plattform für die WWW basierte Maschine-Maschine-Interaktion zu liefern. Entwickelt wurde dieses Framework mit dem Ziel, sich möglichst reibungslos in die bestehenden Infrastrukturen einzufügen [8].

Die größte Herausforderung, die sich für das Framework stellt, ist die starke Heterogenität der bestehenden Systeme. Um dieser zu begegnen, werden im Framework zwei Grundprinzipien forciert: Interoperatibilität und eine einheitliche Repräsentation [8].

Die Interoperatibilität zwischen den Komponenten wird durch die Verwendung des WWW zwar nicht garantiert, zumindest jedoch stark gefördert - schließlich stehen die WWW-Protokolle in nahezu allen Netzwerkumgebungen zur Verfügung. Auf dessen Protokollen aufbauend verwendet das WS-Framework mit dem *Simple Object Access Protocol (SOAP)* [9] einen zwar streng standardisierten, dafür aber auf nahezu allen Systemen verfügbaren Transportmechanismus um XML-Nachrichten zu verschicken, und ermöglicht so eine grundlegende Kommunikation zwischen seinen Systemen [8], [9].

Für die Zusammenarbeit mehrerer Komponenten der WSA reicht es natürlich nicht aus, dass diese über die Möglichkeit verfügen, Nachrichten auszutauschen. Es muss einem Dienstnutzer zudem auch möglich sein, zu einem Dienst in Erfahrung zu bringen, welche Operationen er anbietet, wie er heißt, und in welcher Version er vorliegt. Webservices stellen deshalb grundsätzlich eine in der *Webservices Description Language (WSDL)* [10] verfasste Beschreibung ihrer selbst zur Verfügung. Diese Beschreibung ist unabhängig von den von ihnen verwendeten Protokollen, Interaktionsmodellen oder Programmierkonzepten [8].

Der Vorteil den sich die WSA durch den „on-demand“-Austausch der Schnittstelleninformationen verschafft, ist

der, dass es den Dienstnutzern nun möglich ist, Webservices zur Laufzeit - also „Just-in-time“ - einzubinden, da das Dienstinterface nicht mehr Teil des Quellcodes sein muss [8], [11]. Bleibt man bei dem in der Einleitung gegebenen Beispiel eines Kühlschranks, der sich mit lokalen Supermärkten in Verbindung setzt, so bedeutet dies, dass sowohl neue Supermärkte hinzukommen können als auch alte verschwinden können. Der Kühlschrank ist für seine Funktionalität an keinen bestimmten Supermarkt gebunden, sondern kann sich einfach auf die neue Umgebung einstellen.

### *A. Die WS-Standard Sammlung (WS-*)*

Um den Anspruch einer universellen Architektur zu erfüllen, wurden seit dem Start des WS-Projekts Anfang 2001 eine Reihe von Standards zur Erweiterung der bisher genannten Funktionalität der WSA spezifiziert [12]. In der Literatur wird diese Sammlung oft auch unter dem Kürzel „WS-*“ zusammengefasst. Da der Standardisierungsprozess noch lange nicht abgeschlossen ist, wird an dieser Stelle auf eine vollständige Erfassung aller WS-* Standards verzichtet. Ein - leider nicht mehr topaktueller - Versuch, eine „Webservices Standards Übersicht“ zu geben, kann in [13] nachgelesen werden. Die im Rahmen dieser Arbeit wichtigsten Standards WS-Addressing (III-A.1), WS-Discovery (III-A.2), WS-Transfer (III-A.3), WS-Eventing (III-A.4) und WS-Security (III-A.5) seien im Folgenden kurz vorgestellt:

*1) WS-Addressing:* Das Ziel des WS-Addressing-Standards ist es, Webservices unabhängig von der Vielzahl denkbarer Transportmechanismen einheitlich adressieren zu können. Auf diese Weise soll sichergestellt werden, dass die SOAP-Nachrichten - unabhängig von der physikalischen Netzwerkstruktur - auch zugestellt werden können [14].

Die Informationen, welche nötig sind um einen Dienst oder einen Dienstnutzer - also allgemein einen Nachrichtenendpunkt - zu adressieren, werden im „endpoint reference“-Konstrukt verwaltet. Dieses Konstrukt enthält neben der logischen Adresse des Endpunkts (URI) optional auch noch weitere Informationen - wie z.B. eine SessionID - die zur Kommunikation mit einem Endpunkt benötigt werden [14].

Diese wenigen Informationen würden jedoch nicht ausreichen um beispielsweise eine doppelt empfangene Nachricht zu erkennen. Aus diesem Grund werden in den Nachrichtenkopf noch weitere Informationen wie eine eindeutige „message id“ hinzugefügt [14].

*2) WS-Discovery:* Mit Hilfe des WS-Addressing Standards ist es zwar möglich Nachrichten an Dienste zu schicken, nur ist es dazu zunächst einmal nötig den Dienst zu kennen.

Der klassische Ansatz hierfür ist es, einen zentralen Verzeichnisdienst nach dem gewünschten Dienst zu durchsuchen. Im Umfeld der Webservices wurde für diese Telefonbuchfunktionalität der *Universal Description, Discovery and Integration (UDDI)*-Dienst entwickelt (Abb. 2) [15].

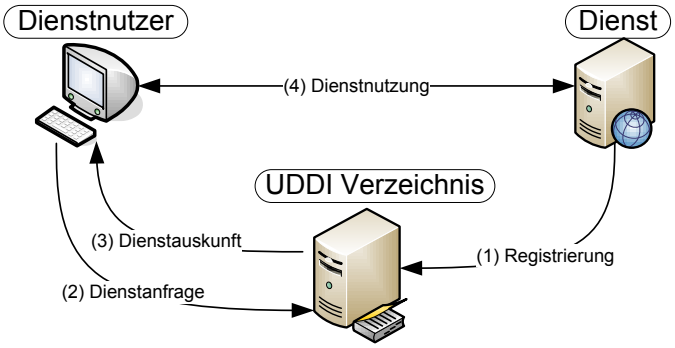



Abbildung 2. Dienstnutzung mit UDDI

Alternativ soll es einem Dienstnutzer jedoch auch möglich sein, seine Dienste dynamisch - insbesondere also ohne Nachfrage bei einem zentralen Verzeichnisdienst - zu finden. Diese Funktionalität wurde im WS-Discovery Standard zusammengefasst [16].

Kernidee dieses Standards ist es, dass alle am Dienstauffindungs-Prozess beteiligten Partner auf an eine spezielle Multicast-Gruppe gesendete öffentliche Nachrichten lauschen.

Es sind nun mehrere Szenarien vorstellbar:

- **„Probe/Probe Match“:**
  Der Dienstnutzer tritt einem bestehendem Netzwerk

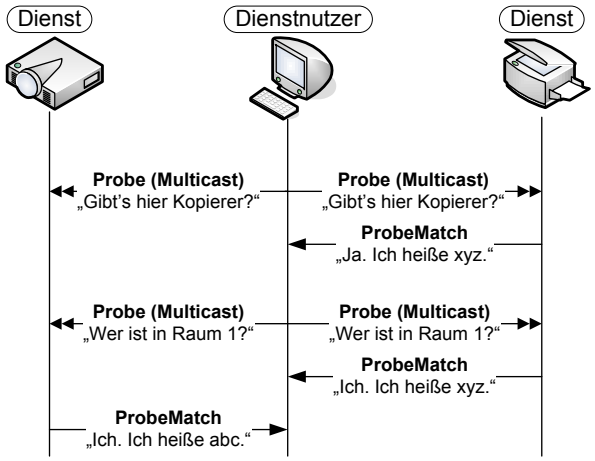



Abbildung 3. Beispiel für Probe/ProbeMatch

bei, in welchem sich potentiell Dienstanbieter befinden, welche die vom Dienstnutzer gesuchte Funktionalität erbringen können. Falls dem so ist, so hat der Nutzer noch nichts über sie in Erfahrung gebracht. Er kennt also beispielsweise lediglich den Typ des von ihm gewünschten Dienstes - nicht jedoch seine logische Adresse [16].
Um nun die logischen Adressen derjenigen Dienste in Erfahrung zu bringen, welche z.B. auf eine vorliegende Typbeschreibung passen, schickt der Dienstnutzer

eine „Probe“-Nachricht mit allem was er über seinen Zieldienst weiß an die gemeinsame Multicast-Adresse. Alle Dienste, deren Profil auf die in der „Probe“-Nachricht angegebenen Kriterien passt, generieren daraufhin als Antwortnachricht eine „Probe Match“-Nachricht, in welcher sie ihre eigene „endpoint reference“ an den anfragenden Endpunkt verschicken (Abb. 3) [16].

- **„Hello/Bye“:**
  Ebenso wie der Fall, dass der Dienst *vor* dem Dienstnutzer Teil des Netzwerkes ist, kann das genaue Gegenteil auftreten. Damit die Dienstnutzer nicht immer wieder anfragen müssen, ob ein gewünschter Dienst (noch) zur Verfügung steht, versenden Dienste nachdem sie einem Netzwerk beigetreten sind eine „Hello“-Nachricht und kurz bevor sie ein Netzwerk verlassen eine „Bye“-Nachricht an die Multicast-Gruppe (Abb. 4). Während die „Bye“-Nachricht lediglich die

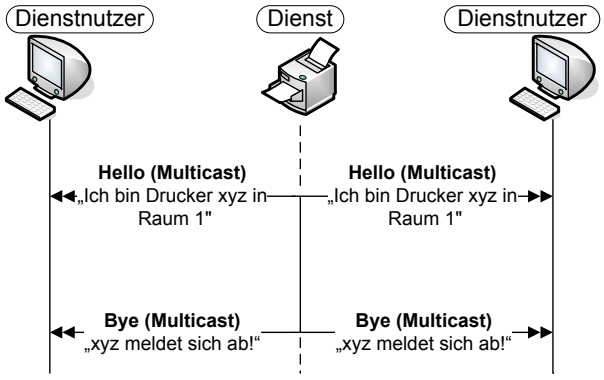



Abbildung 4. Beispiel für Hello/Bye

„endpoint reference“ des versendenden Dienstes enthält, werden in der „Hello“-Nachricht zusätzlich noch weitere Informationen über den Dienst - wie z.B. sein Typ - mitgeschickt [16].
Geht man davon aus, dass alle verschickten Nachrichten ihr Ziel auch erreichen, so ist es einem Dienstnutzer also möglich, eine vollständige und korrekte Liste aller verfügbaren Dienste innerhalb des Netzwerkes zu führen. Er muss dazu lediglich nach einer einzigen „Probe“-Anfrage kurz nach dem Betreten des Netzwerks auf eingehende „Hello“- und „Bye“-Nachrichten lauschen [16].

- **„Resolve/Resolve Match“:**
  Kennt der Dienstnutzer bereits die logische Adresse seines Wunschdienstes (verfügt also über dessen „endpoint reference“), benötigt er nun noch die dazu passende auf Netzwerkebene auflösbare Transportadresse. Um letztere zu erhalten, schickt er dem WS-Discovery Standard konform eine sog. „Resolve“-Multicast-Nachricht (Abb. 5). Diese Nachricht enthält die abstrakte „endpoint reference“ und wird von dem gesuchten Dienst mit einer Unicast „Resolve Match“-

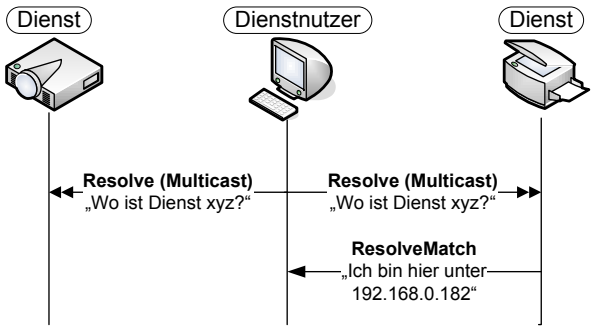



Abbildung 5. Beispiel für Resolve/ResolveMatch

Nachricht beantwortet. Die Antwortnachricht enthält nun die als XAddr bezeichneten Transportadressen, unter denen der Dienst angesprochen werden kann [16].

Stellt man sich nun ein Netzwerk mit einer sehr großen Anzahl oder einer hohen Fluktuation von Dienstnutzern vor, so folgt daraus, dass sehr viele Multicast-Nachrichten versendet werden. Wenn jede dieser Nachrichten wie gewünscht bei allen Empfängern ankommt (was zu Stoßzeiten durchaus unrealistisch ist), so bedeutet dies, dass jeder Nachrichtenendpunkt diese Nachricht verarbeiten muss - auch wenn sie nicht für ihn gedacht ist. Auf diese Weise können gerade eingebettete Systeme sehr schnell an ihre Leistungsgrenzen stoßen.

Aus diesem Grund wurde im WS-Discovery Standard ein sogenannter *Discovery Proxy (DP)* vorgesehen. Die Aufgabe des DPs besteht darin, das Aufkommen von Multicast-Nachrichten auf ein Minimum zu beschränken und den Dienstauffindungsprozess als solchen zu beschleunigen [16].

Wie bereits oben beschrieben, ist es einem Teilnehmer des Netzwerkes - in diesem Fall dem DP - möglich, eine vollständige Liste aller innerhalb des Netzwerks verfügbaren Webservices zu führen, ohne dazu aktiv ins Geschehen eingreifen zu müssen. Dieses Wissen nutzt der DP nun dazu, anderen Dienstnutzern das Versenden von Multicast-Nachrichten zu ersparen [16]:

Auf jede von einem Nutzer ausgehende „Probe“ oder „Resolve“ Nachricht antwortet ein DP sofort mit einer „Hello“ Nachricht. Nachdem der Nutzer so von der Existenz des DPs in Kenntnis gesetzt wurde, schickt er alle weiteren „Probe“ oder „Resolve“ Nachrichten von nun an direkt an den DP [16].

*3) WS-Transfer:* Der WS-Transfer Standard stellt grundlegende Basisoperationen zur Interaktion mit Ressourcen zur Verfügung. Ähnlich wie beim HTTP-Protokoll werden die Operationen „GET“, „PUT“ und „DELETE“ spezifiziert [17]:

- **„GET“:** Liefert eine Momentaufnahme einer Ressource.
- **„PUT“:** Ersetzt eine alte Repräsentation einer Ressource durch eine neue.

- **„DELETE“:** Löscht eine Ressource gänzlich.

*4) WS-Eventing:* Die meisten Menschen wollen gerne auf dem neuesten Stand der Dinge gehalten werden. Da es etwas umständlich ist, ständig nachzufragen, ob sich etwas geändert hat („Sind wir schon da?“), wird es in der Realität meist so gehandhabt, dass derjenige, der etwas gemacht hat, alle die es interessiert über diese Aktion informiert.

Das Pendant zu diesem Anwendungsgebiet wird innerhalb der WSA durch den WS-Eventing Standard [18] abgedeckt. Einem Dienstnutzer ist es unter Verwendung dieses Standards möglich, sich bei einem Dienst für den Empfang bestimmter Nachrichten anzumelden (engl. to subscribe). Diese Anmeldung kann zeitlich begrenzt oder unbegrenzt sowie für alle Ereignisse oder nur für bestimmte erfolgen. Des weiteren wird das Konzept eines Stellvertreters unterstützt, bei dessen Benutzung die Anmeldung nicht direkt am System, in dem das Ereignis auftritt, vorgenommen wird, sondern von diesem an einen externen Dienst verwiesen wird.

Solange ein Dienstnutzer für den Erhalt von Benachrichtigungen über das Eintreten eines bestimmten Ereignisses registriert ist, werden bei jedem Auftreten dieses Ereignisses Nachrichten an ihn gesendet [18].

*5) WS-Security:* Obwohl die bisher genannten WS-* Standards die Grundlage fast aller Webservices stellen, lassen sie doch alle die Frage nach einem systemübergreifenden Sicherheitssystem offen. Diese Lücke wird vom WS-Security Standard geschlossen [19].

Um für die SOAP-Nachrichten die Schutzziele Integrität, Vertraulichkeit, Verbindlichkeit und Authentizität [20] zu garantieren, ist der einfachste Ansatz eine wechselseitig authentifizierte sichere Punkt-zu-Punkt-Verbindung zwischen den Endpunkten aufzubauen. Ein Beispiel für eine solche Verbindung könnte ein *Transport Layer Security (TLS)* [21] Kanal sein (Abb. 6).

Es sind jedoch Anwendungsfälle denkbar, in denen dieser Ansatz keine Alternative darstellt:

- Kommunikation mit einer unbekannten Anzahl von Partnern (Multicast-Nachrichten).
- Es besteht keine direkte Verbindung zwischen den Kommunikationspartnern.

Der WS-Security Standard widmet sich deshalb der Frage, wie die Integrität und die Vertraulichkeit von in nicht vertrauenswürdigen Umgebungen versendeten SOAP-Nachrichten auf Nachrichtenebene gewährleistet werden kann. Zu diesem Zweck wird auf die beiden Standards „XML Encryption“ [22] und „XML Signature“ [23] zurückgegriffen um die SOAP-Nachrichten zu verschlüsseln bzw. zu signieren. Zudem wird im Standard definiert, wie Sicherheits-Tokens - wie z.B. X.509 Zertifikate [24] - als Teil einer SOAP-Nachricht übertragen werden, um so die Authentizität der Identität eines Kommunikationspartners überprüfen zu können [19].

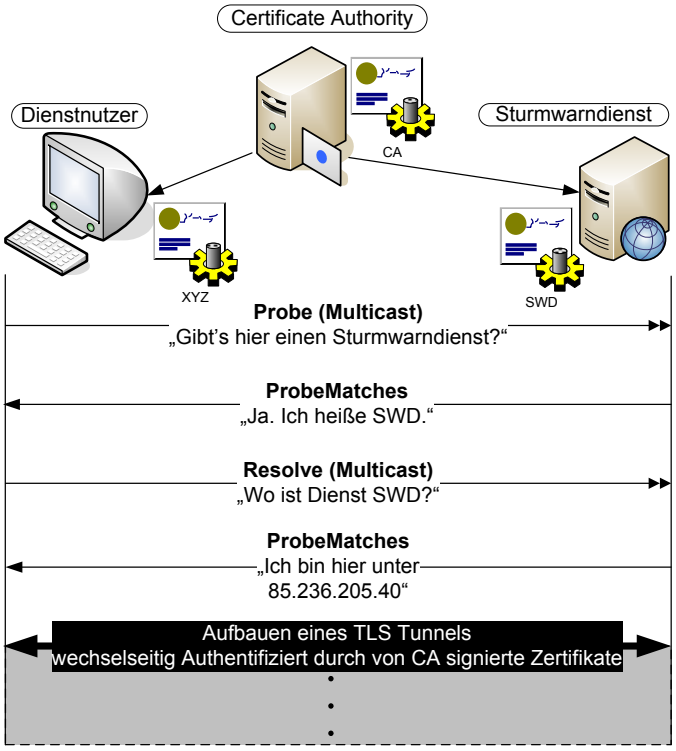



Abbildung 6. Dienstauffindung und sichere Kommunikation durch TLS Kanal

### B. Profile

Als „Profil“ wird eine Sammlung von Webservice Spezifikationen in Kombination mit einer Konvention, welche das Zusammenspiel der einzelnen Standards regelt, bezeichnet. Profile wurden von der „Web-Services Interoperability Organization“ (WS-I) eingeführt um die Interoperatibilität zwischen verschiedenen Webservices zu vereinfachen [25].

Neben dem „Basic Profile“ wurden eine Reihe weiterer Profile wie das „Attachments Profile“, das „Basic Security Profile“ oder auch das im Folgenden näher beschriebene „Devices Profile for Web-Services“ veröffentlicht [13].

## IV. Das Devices Profile

Die Zusammenarbeit der Parteien, die an einer Anpassung der WSA für eingebettete Systeme interessiert waren, führte zur Spezifizierung des *Devices Profile for Web Services (DPWS)*. Dieses wurde im Hinblick auf eingebettete Systeme als ein Web-Service Profil entworfen, welches zwar grundlegende Web-Service Funktionalitäten ermöglicht, zugleich jedoch nur minimale Anforderungen an die zur Verfügung stehenden Ressourcen stellt [26]. Unterstützt werden sollen [26]:

- Das sichere Versenden von Nachrichten (Abschnitte IV-B und IV-F)
- Dynamische Dienstauffindung (Abschnitt IV-C)
- Einheitliche Beschreibung der Dienste (Abschnitt IV-D)
- Ereignisbasierte Kommunikation (Abschnitt IV-E)

Im Rahmen dieser Spezifikation wurde nun eine minimale Menge von WS-Standards so modifiziert und zusammengefasst, dass diese Unterstützung auch von eingebetteten Systeme möglich ist. Federführend beteiligt sind hierbei die in III-A vorgestellten WS-* Standards [26].

### A. Client, Hosted Service und Device

In der Terminologie des DPWS existieren zwei unterschiedliche Netzwerkendpunkte: der Dienst (engl. Service) und der Dienstnutzer (engl. Client). Während der Client analog zum WS-Adressing-Standard (Kap. III-A.1) schlicht als Nutzer eines Dienstes verstanden wird, ist der Begriff Service weiter unterteilt in Gerät (engl. Device) und gehosteter Dienst (engl. Hosted Service) [26].

Der Unterschied zwischen Device und Hosted Service lässt sich am besten an einem Beispiel verdeutlichen. Angenommen innerhalb eines Netzwerkes würde ein Multifunktionsgerät seine Dienste Drucken, Faxen und Energiesparen über das DPWS zur Verfügung stellen. In diesem Fall wäre der Drucker im Sinne der DPWS-Terminologie ein Device, die Dienste Drucken, Faxen, Energiesparen wären Hosted Services (Abb. 7).

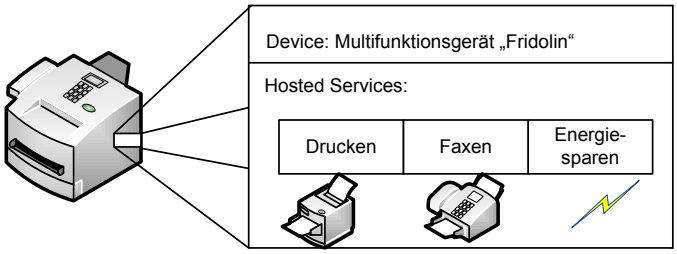



Abbildung 7. Beispiel eines Devices mit drei Hosted Services

Wie das Beispiel zeigt, können auf einem Device mehrere Hosted Services zur Benutzung angeboten werden - ein Hosted Service befindet sich jedoch immer auf einem einzigen Device [26].

### B. Nachrichtenvermittlung

DPWS baut auf dem in Kapitel III-A.1 vorgestellten WS-Adressing-Standard auf, um eine Nachrichtenzustellung zu ermöglichen. Da es dem DPWS jedoch im Gegensatz zu dieser Vorlage nicht darum geht, die Türen für unterschiedlichste zukünftige Erweiterungen offen zu halten, schränkt das Devices Profile den WS-Adressing-Standard an vielen Stellen ein. Zudem werden Abstriche an den Implementierungen von URI, HTTP, SOAP und weiteren Protokollen zugelassen. So wird beispielsweise nicht gefordert, dass DPWS konforme Geräte SOAP Nachrichten größer als 32,767 Byte verarbeiten können müssen [26].

### C. Dienstauffindung

In Kapitel III-A.2 wurde bereits dargelegt, nach welchen Schemata die Dienstauffindung innerhalb des WS-Frameworks erfolgen kann. In der Einsatzumgebung eingebetteter Systeme kann jedoch nicht davon ausgegangen werden, dass der Zugriff auf einen zentralen Verzeichnisdienst jederzeit möglich ist. Aus diesem Grund verwendet das DPWS die im WS-Discovery Standard spezifizierten dynamischen Dienstauffindungsmethoden. Der in diesem Standard verwendete „Dienst“-Begriff wird jedoch genauer spezifiziert.

Wie bereits in Kapitel IV-A ausgeführt, teilt das DPWS Dienste in die Typen „Device“ und „Hosted Service“ ein. Um die Anzahl der sich am Dienstauffindungsprozess beteiligenden Partner möglichst gering zu halten, werden innerhalb des Profils nur Devices als auffindbare Dienste im Sinne des WS-Discovery Standards betrachtet. Ziel dieser Beschränkung ist es, den durchschnittlichen Netzwerkverkehr zu reduzieren und so für eingebettete Umgebungen attraktiver zu werden [26].

Kombiniert man diese Technik mit dem in Kapitel III-A.2 vorgestellten Discovery Proxy, so lassen sich die Anforderungen an die Netzwerkressourcen noch weiter reduzieren.

### D. Einheitliche Beschreibung der Dienste

Wie bereits in Kapitel III diskutiert, garantiert das WS-Framework, dass für alle verfügbaren Dienste eine in WSDL gehaltene Beschreibung ihrer Schnittstelle abrufbar ist. Diese Tatsache ermöglicht es auch Dienste zu verwenden, deren Interface zur Programmerstellungszeit noch nicht bekannt war.

Nachdem ein Device als solches keine dem Client nützliche Funktion erbringt, hat letzterer also natürlich vorrangig ein Interesse daran, die Dienstbeschreibungen der Hosted Services jenes Devices anzufordern.

Zu diesem Zweck sendet ein Client dem DPWS entsprechend eine WS-Transfer konforme „GET“-Anfrage (vgl. Kap. III-A.3) an das Device und erhält als Antwort eine Nachricht mit den Kenndaten des Geräts zusammen mit der Angabe der von ihm gehosteten Dienste und deren Adressen. Mit Hilfe dieser Adressen kann der Client nun jeweils eine „GET“-Anfrage an die Hosted-Services senden. Eine Antwort auf eine dieser Anfragen enthält neben einigen Kenndaten der Dienste die gesuchte Schnittstellenbeschreibung oder eine URL unter welche diese zu finden ist (Abb. 8) [26].

### E. Ereignisbasierte Kommunikation

Ein mögliches Einsatzgebiet für eingebettete Systeme ist das eines Sensornetzwerkes. In [11] wird beispielsweise der Aufbau eines auf DPWS basierten Tracking Systems für Bluetooth Geräte vorgestellt. Das System besteht aus einer Reihe verteilter Sensorsysteme, welche sowohl periodisch, als auch beim Entdecken eines Bluetooth Gerätes Nachrichten an einen zentralen Knotenpunkt senden. Um diese Ereignis basierte Kommunikation zu ermöglichen greift das DPWS auf den WS-Eventing Standard (vgl. Kap. III-A.4) zurück [18], [26].

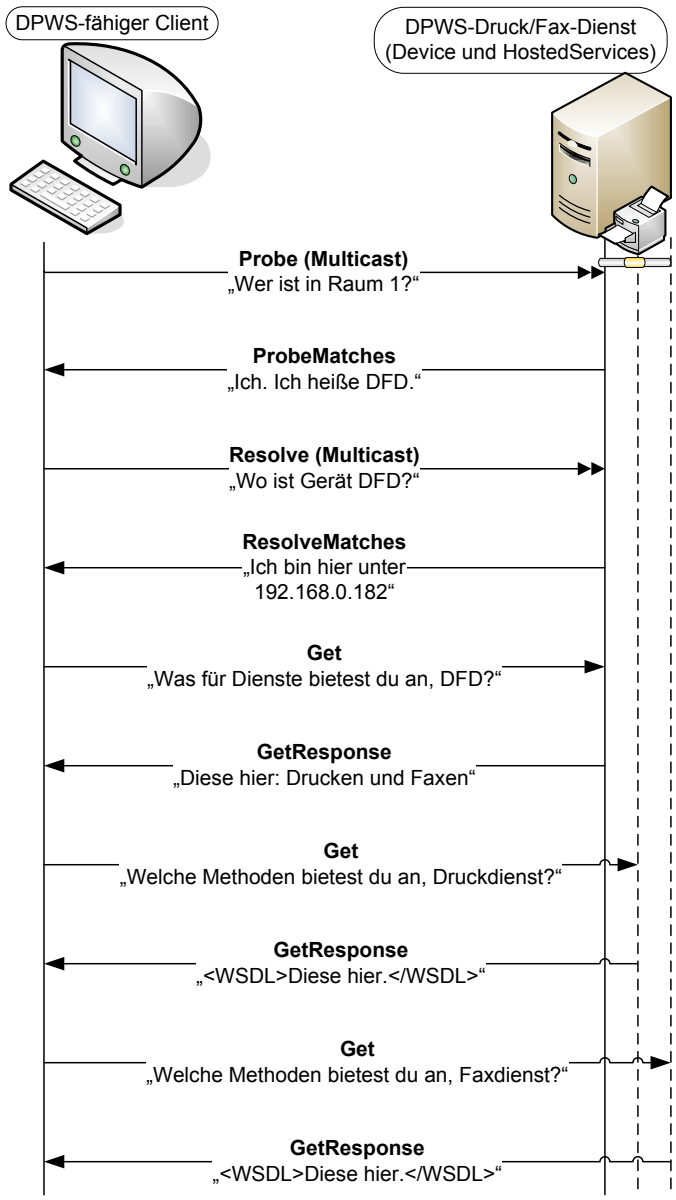



Abbildung 8. DPWS: Transfer der Diensteschreibungen

### *F. Sicherheit*

Wie bereits in Kapitel III-A.5 diskutiert, werden innerhalb des WS-Frameworks zwei unterschiedliche Techniken verwendet, um die Schutzziele [20] Integrität, Vertraulichkeit, Verbindlichkeit und Authentizität zu erreichen: Einsatz kryptographischer Verfahren auf Nachrichtenebene und Kommunikation durch einen sicheren Kanal. Innerhalb des DPWS gilt folgende Empfehlung [26]:

WS-Discovery Nachrichten im Sinne von Kapitel III-A.2 sollten auf Nachrichtenebene signiert werden, um so ihre Integrität und die Überprüfbarkeit der Authentizität der Senderidentität sicherzustellen. Da die WS-Discovery-Nachrichten dafür gedacht sind, von jedem Gerät innerhalb eines Netzwerkes gelesen werden zu können, wird auf eine Verschlüsselung und somit auf eine Nachrichtenvertraulichkeit verzichtet [26].

Die übrigen Nachrichten werden im DPWS separat behandelt. Da deren Übertragung stets von *einer* Quelle zu *einem* Empfänger stattfindet, sollten sie stets über einen gesicherten Kanal versendet werden. Auf diese Weise können gleichzeitig alle geforderten Schutzziele erfüllt werden [26].

## V. Verwandte Arbeiten

Das DPWS entstand zwar aus der Kooperation Namhafter Firmen wie Microsoft, Intel und Lexmark, konnte jedoch natürlich nicht verhindern, dass weitere Parteien eigene Ansätze für Webservices auf eingebetteten Systemen realisierten. Diese alternativen Ansätze, wie der in [27] speziell für PDAs konzipierte, oder der in [28] für bargeldloses Bezahlen entwickelte, sind im Vergleich zum DPWS jedoch wesentlich weniger ausgereift. Meist existiert zwar eine Pilotimplementierung, eine offizielle Spezifikation oder ein Entwicklungsframework für eigene Anwendungen wurden jedoch nur für das DPWS veröffentlicht.

Es stellt sich natürlich auch die Frage, ob die Verwendung von Web-Services überhaupt die geeignetste Lösung ist, um Nutzer mit Diensten interagieren zu lassen. Mit dem *Universal Plug and Play (UPnP)* [29] existiert bereits eine für den Heimnetzwerkbereich entwickelte SOA, welche es Nutzern ebenfalls erlaubt, automatisch Dienste zu finden und einfach zu konfigurieren. Im direkten Vergleich mit einer auf DPWS basierenden Lösung kann UPnP jedoch nur wenig punkten:

Den Autoren in [30], [31] zufolge hat UPnP Probleme mit der Skalierbarkeit des Dienstauffindungsprozesses. Dies verwundert nicht weiter, da UPnP weder das Konzept des Discovery Proxys noch die Beschränkung der Dienstauffindung auf Geräte betreibt. Des weiteren ist im Gegensatz zum DPWS die Sicherheitsfrage in der UPnP-Spezifikation selbst ungeklärt, was zu einer inhomogenen Sicherheitslandschaft führte [32], [33]. Ein weiterer Punkt, der für das DPWS spricht, ist die Tatsache, dass DPWS-fähige Geräte nach wie vor von Komponenten der uneingeschränkten WSA genutzt werden können. UPnP hingegen ist ein eigenes Konzept und lässt sich nicht so leicht in komplexe Geschäftsprozesse einbinden. Letztendlich sei hier noch angeführt, dass im Gegensatz zum UPnP beim DPWS die Dienstauffindung leicht über die Grenzen der lokalen Broadcast Domain hinweg ausgedehnt werden kann.

Weitere Lösungen für die einfache Zusammenarbeit innerhalb eines verteilten Systems wurden beispielsweise mit der *Home Audio Video Interoperability (HAVi)* [34] Technologie speziell für den Multimediaeinsatz oder der Jini SOA Implementierung für Java [35] entwickelt. Beide Lösungen sind jedoch auf die Verwendung spezieller Hardware bzw. Software angewiesen und eignen sich deshalb weniger für universelle systemübergreifende Anwendungen.

## VI. Zusammenfassung und Ausblick

Wie diese Arbeit gezeigt hat, ist es mit dem bereits im Konzept der Web-Services vorhandenen Ansatz der Profilbildung relativ einfach möglich, die WSA so anzupassen, dass sie auf eingebetteten Systemen zum Einsatz kommen kann. Diese Anpassung - auf den Namen DPWS getauft - kann derartig vorgenommen werden, dass zentrale Funktionalitäten der WSA nach wie vor möglich sind, auch wenn an vielen Stellen die Universalität der Ressourceneffizienz weichen musste. Trotz dieser Beschränkungen ist der Ansatz, eingebettete Systeme unter Verwendung von Web-Servces zu vernetzen, anderen Technologien wie dem UPnP überlegen.

Obwohl das DPWS eine relativ junge Technologie ist, wurden bereits eine Vielzahl mehr oder weniger ausgereifter Implementierungen entwickelt. Neben den quelloffenen Varianten wie dem *WS4D-gSOAP Stack* [36] für C/C++ oder dem *WS4D Java Multi Edition DPWS Stack (JMEDS)* [36] existiert mit der *Web Service on Devices API (WSDAPI)* auch eine Implementierung für die Microsoft Betriebssysteme *Windows Vista* und *Windows Server 2008* [37].

Diese Kombination von Open-Source und Marktführer-Software lässt auf eine rasche Verbreitung und einen breiten Einsatz der Technologie bereits in naher Zukunft hoffen.

## Literatur


[1] (2009, Mai) http://www.webservicelist.com/.

[2] I. Jacobs, "Architecture of the world wide web, volume one," W3C, Tech. Rep., Dezember 2004, http://www.w3.org/TR/webarch/.

[3] T. Bray *et al.*, "Extensible markup language (xml) 1.0 (fourth edition)," W3C, Tech. Rep., August 2006, http://www.w3.org/TR/2006/REC-xml-20060816/.

[4] T. Berners-Lee *et al.*, *Hypertext Transfer Protocol – HTTP/1.0*, Network Working Group Std., Mai 1996, http://tools.ietf.org/html/rfc1945.

[5] S. Hashimi. (2003, August) Service-oriented architecture explained. http://www.ondotnet.com/pub/a/dotnet/2003/08/18/soa_explained.html.

[6] M. Schoeberl. (2006) Embedded systems. TU-Wien. http://www.jopdesign.com/teaching/EmbeddedIntro.pdf.

[7] M. M. Group. (2009, Juni) http://www.internetworldstats.com/stats.htm.

[8] F. Curbera *et al.*, "Web services: Why and how," August 2001, http://lsdis.cs.uga.edu/courses/8351Spring2006/papers2Read/wsWhyandHow.pdf.

[9] M. Gudgin *et al.*, "Soap version 1.2 part 1: Messaging framework (second edition)," W3C, Tech. Rep., April 2007, http://www.w3.org/TR/soap12-part1/.

[10] E. Christensen *et al.*, "Web services description language (wsdl) 1.1," W3C, Tech. Rep., März 2001, http://www.w3.org/TR/wsdl.html.

[11] A. Bobek, "Serviceorientierte infrastrukturen für vernetzte dienste und eingebettete geräte," Dissertation, Fakultät für Informatik und Elektrotechnik der Universität Rostock, Juli 2008, http://rosdok.uni-rostock.de/file/rosdok_derivate_000000003716/Dissertation_Bobek_2009.pdf?hosts=local.

[12] W. W. W. Consortium. Web services activity. w3. http://www.w3.org/2002/ws/.

[13] (2007, Februar) Web services standards overview. innoQ. http://www.innoq.com/soa/ws-standards/poster/.

[14] D. Box *et al.*, "Web services addressing (ws-addressing)," W3C, Tech. Rep., August 2004, http://www.w3.org/Submission/ws-addressing/.

[15] T. Bellwood *et al.*, *UDDI Spec Technical Committee Draft*, OASIS Std., Rev. 3.0.2, Oktober 2004, http://uddi.org/pubs/uddi_v3.htm.

[16] J. Beatty *et al.*, "Web services dynamic discovery," W3C, Tech. Rep., April 2005, http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf.

[17] J. Alexander *et al.*, "Web services transfer (ws-transfer)," W3C, Tech. Rep., September 2006, http://www.w3.org/Submission/WS-Transfer/.

[18] D. Box *et al.*, "Web services eventing (ws-eventing)," W3C, Tech. Rep., März 2006, http://www.w3.org/Submission/WS-Eventing/.

[19] A. Nadalin *et al.*, "Web services security: Soap message security 1.1 (ws-security 2004)," OASIS, Tech. Rep., Februar 2006, http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf.

[20] C. Eckert, *IT-Sicherheit: Konzepte-Verfahren-Protokolle*, 5th ed. Oldenbourg, 2008.

[21] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol - Version 1.1*, Network Working Group Std., April 2006, http://tools.ietf.org/html/rfc4346.

[22] T. Imamura *et al.*, "Xml encryption syntax and processing," W3C, Tech. Rep., Dezember 2002, http://www.w3.org/TR/xmlenc-core/.

[23] M. Bartel *et al.*, "Xml signature syntax and processing (second edition)," W3C, Tech. Rep., Juni 2008, http://www.w3.org/TR/xmldsig-core/.

[24] D. Cooper *et al.*, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, Network Working Group Std., Mai 2008, http://tools.ietf.org/html/rfc5280.

[25] WS-I. (2002, Februar) Web service profiles - an introduction. IBM Corporation, Microsoft Corporation. http://www.ws-i.org/docs/WS-I_Profiles.pdf.

[26] S. Chan *et al.*, "Devices profile for web services," XMLSOAP, Tech. Rep., Februar 2006, http://specs.xmlsoap.org/ws/2006/02/devprof/devicesprofile.pdf.

[27] M. Asif, S. Majumdar, and R. Dragnea, "Hosting web services on resource constrained devices," in *Web Services, 2007. ICWS 2007. IEEE International Conference on*, Salt Lake City, UT„ Jul. 2007, pp. 583–590.

[28] *Web Services on Mobile Devices - Implementation and Experience*, IEEE Computer Society. 5th IEEE Workshop on Mobile Computing Systems and Applications, 2003.

[29] A. Presser, L. Farrell, D. Kemp, W. Lupton *et al.*, "Upnp device architecture," UPnP Forum, Tech. Rep., 2008, http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.

[30] E. Zeeb, F. Golatowski, and D. Timmermann, "Web services - zu groß für eingebettete systeme ?" in *12th Symposium Maritime Elektrotechnik, Elektronik und Informationstechnik*, Oktober 2007.

[31] S. Cheshire, "How does zeroconf compare with viiv/dlna/dhwg/upnp?" Zeroconf, http://www.zeroconf.org/ZeroconfAndUPnP.html.

[32] Unknown, "Devicesecurity:1 service template," UPnP Forum, Tech. Rep., November 2003, http://www.upnp.org/standardizeddcps/documents/DeviceSecurity_1.0cc_001.pdf.

[33] A. Klemets *et al.*, "Upnp authentication and authorization," USA Patent 20 080 092 211, April, 2008, http://www.freepatentsonline.com/20080092211.html.

[34] C. Members. About havi objectives and about havi overview. HAVi. http://www.havi.org/about/.

[35] (2006, August) Category:introduction to jini. Jini. http://www.jini.org/wiki/Discovery.

[36] F. Golatowski, A. Bobek, and E. Zeeb. (2007-2009) Web services for devices. ws4d. http://www.ws4d.org/.

[37] Microsoft. (2009) About web services on devices. http://msdn.microsoft.com/en-us/library/aa385800(VS.85).aspx.

# Delay Tolerant Networks

Manuel Scharf
Betreuer: Tobias Bandh
Seminar Innovative Internettechnologien und Mobilkommunikation SS 2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: scharfm@in.tum.de

## Abstract
DTN is an approach to develop an architecture for heterogenous networks with a connection suffering from long signal propagation delay and disruption. Some requirements for applications in the DTN sector will be shown, followed by an analysis of problems when using the common protocols of the Internet. This is followed by an overview of how the DTN architecture, which is developed by the Delay-Tolerant Network Research Group (DTNRG), handles these challenges.

## Keywords
Delay disruption network, DTN, space communication, interplanetary Internet, protocol, solar planetary network, network standard

## 1. Introduction
Delay and disruption affects every communication among electronic devices. The time it takes until a signal arrives at the receiver is affected by the distance of the communication partners and by the velocity of signal propagation. As long as this time is short it does not have a big influence on communication. But there are cases when this time becomes significant. A demand for communication with long delay links first emerged in space science. With the further success of communication networks and its spreading to the space and remote locations, a solution had to be found to handle this conditions. This is necessary as current networking architectures are not able to support long delay links properly, this will be explained in more detail in Section 3. The main responsible group for research and standardisation concerning DTN is DTNRG. There isn't a reference implementation yet, but some successful tests for space networks have been carried out and parts of the DTN architecture are already in use in some applications. That means in addition, that there are still some unresolved problems concerning DTN which require further research.

## 2. Networks with special requirements
As mentioned before there is always some delay in communication. But in the subsequently mentioned examples delayed, intermittent, disrupted or disturbed data transfers play a central role. Space networks are the most obvious example of networks with delay, but the methods of handling this delay are also useful in other applications.

### 2.1 Space networks
Since the first satellite Sputnik-1, from the year 1957, the action of human being hasn't been limited to earth. Diverse space missions have different communication requirements. There are different types of communication such as normal communication to humans in space stations and shuttles. Others include transfer of mission data like photographs, height scanning and other measured and recorded data and control of course and other operational parameters. These require a high bandwidth. The connection is obviously wireless by radio or radiated laser light. Thus communication can be interrupted either in predictable intervals, due to the planets rotation or orbital motion, or unexpectedly, by space dust or solar flare and its radiation.

A particular characteristic of space communication is the extremely long distance between two communication partners. The fact of a limited speed of light results in high signal propagation delays from parts of a second from an earths satellite to minutes to the inner planets and hours to the outer planets and probably even more for projects in future.

### 2.2 Sensor Networks
According to [1] a wireless sensor network consists of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions. The devices are usually small and powered by a battery. They communicate via a radio transmitter and take their measurements at fixed intervals. To reduce energy consumption and get a longer operation time the device is in sleeping mode most of the time. Has a limited range and only takes measurements and switches on the radio transmitter for a short time. They might be attached to objects in motion. These sensor devices usually build up an ad hoc wireless network for communication and transfer their data with the help of other sensors acting as relays to a base station. Sensor networks, with devices which are sometimes connected to one another, can profit from a delay tolerant network like space networks.

### 2.3 Terrestrial Wireless Networks
There are districts in the world where a permanent Internet connection is not always available or not available at all. The poor or lost connection can be caused by limited electrical power, systems prone to failure due to pervasive dust and lack of maintenance, high repair expenses and lack of spare parts. In this districts it might be favorable that a data transfer is intermittent or delayed but doesn't fail. In the case of a non-connected client at a rural home, the data to be sent could be forwarded to a mobile device. When the mobile device travels to another location where it gets network connection, the data can be further relayed. The fact that a continuous end to end connection does not or not always exist is shared with the other examples and makes a network architecture, like DTN, which addresses this fact, useful in this scope.

### 2.4 Underwater Communication
Water has a very high attenuation of radio waves, which requires extremely low frequencies (ELF). But low frequencies require long antennas, which are almost impossible to use. Usually

submarines surface or get close to the water surface for radio communication. Acoustic communication has been proven to work much better under water. This technique is used for diver conversation and in scientific applications for underwater sensor communication. In addition to the aspects of the terrestrial sensor networks, underwater networks suffer a reasonable propagation delay, limited bandwidth and reasonable disturbances. These problems are caused by different noise sources, by a self eliminating effect of the sound wave due to its reflection and phase shift at the water surface and by different propagation speeds at different temperatures. This makes well adapted network architecture much more necessary than in common sensor networks.

## 2.5 Opportunistic Contacts

There are many cases where a network connection is not always available or is too expensive to be used. In these cases it could be useful if the data would be transferred to other devices which might get earlier access to the desired recipient. Or a large amount of data could be sent when a cheap and broadband connection is available. For example WLAN or wired link instead of GSM data connection. Important in this case is that a data transfer does not fail, but is just delayed and resumes when possible without user interaction. Security aspects must be followed when using foreign networks or devices as forwarding stations. Again this store and forward procedure resembles the characteristics of a DTN.

Figure 1 which was inspired by [2] shows a combination of the previously mentioned examples. There is a changing and intermittent connection between mobile and fixed devices, sensors, aircrafts, space stations, satellites to a global or solar network. For security reasons some devices should not get access to all devices in the network, but in principle there should be no technical reason preventing a connection.

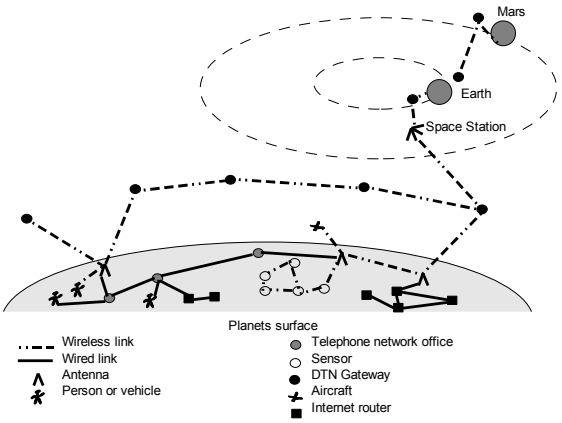



Figure 1: A possible delay tolerant network [2]

## 2.6 Common Application Characteristics

The applications mentioned in the previous sections looks very different at first glance, but when the requirements which enable such applications are analyzed the following characteristics can be derived.

- High propagation delay or no constant connection
  - No immediate forward possible, store and forward useful
  - No instant feedback possible, avoid many message exchanges
- High error rates
  - Perform error detection at each possible forwarding device
  - Use advanced error detection
- Heterogenous devices and protocols
  - Use overlay to abstract differences in lower layers
- Flexible network
  - Communication parameters change during a period of time (bandwidth, location, connection)
  - Naming and routing parameters for different situations
- Use resources in an economic way
  - Available bandwidth
  - Use the available time of connection intervals effectively
  - Device specific parameters like storage, CPU-time and energy
- High security demands
  - Proper authentication without immediate access to a central server and detection of compromised communication partners
  - Authorization with support of different privileges depending on time, bandwidth, total data amount
  - Fastest and earliest possible reactions (in terms of the forwarding chain) on security concerns

# 3. Current network architectures and limitations

Before getting a closer look at the properties of the DTN it is useful to check existing protocols for their suitability for the mentioned application characteristics.

## 3.1 Internet Protocol Family

By building the backbone for Internet communication the protocols TCP and IP have proven to be very powerful. Using TCP and IP would have many advantages. It reduces the development effort and testing and leads to lower costs for the whole application.

### *3.1.1 IP*

At the layer of the Internet-Protocol the use of IPv4 might be suggested. But with the connection of some DTN's to the Internet the problem of almost exhausted addresses is made worse and will not be a good start for a growing space network. IPv6 should be used for new DTN deployments but with the consequence of no smooth connection to the Internet without tunneling. The Internet-Protocols are connectionless and provide only best effort delivery, therefore IP with its low complexity is usable for DTN but reliability must be provided by upper layer protocols. Routing decisions have to be adapted to get a good DTN performance.

### *3.1.2 TCP*

The transfer control protocol is a connection oriented protocol and is therefore very conversational. This means that several messages are necessary between both communication partners before a connection is established and data can be sent (Figure 2). When using TCP in its most common versions on the Internet with congestion control Tahoe, Reno or NewReno, a proper TCP

operation suffers from several characteristics mentioned in 2.6 as described in [2].

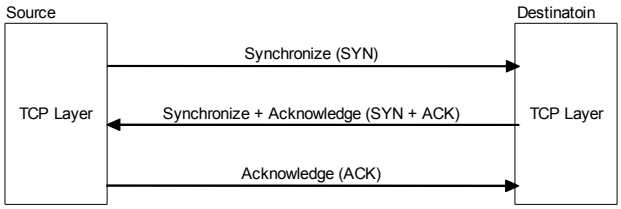



Figure 2: Three way handshake for connection establishment in TCP

An end to end path between sender and receiver is necessary to build up an TCP connection, but this is not possible in all applications.
TCP does not perform well if a low signal to noise ratio or if disruptions cause high error rates. This is caused by the fact that error detection is made by the final receiver. For example a landing module at the Saturn moon Titan wants to send collected data to the earth. The data will first be sent to an orbiting sonde of Saturn, later forwarded to a repeater located half way to earth and finally forwarded to earth. The accumulated time delay would be approximately 1.5 hours. If there is a signal error during data transfer from Titan to the Saturn's sonde (likely due to the dense atmosphere) the erroneous data would be forwarded to the Earth nevertheless. It would take a whole round trip time of 3 hours until the titan lander starts to resend the data. Should there be an error detection at each forwarding device including the Saturn's sonde, an error would add only the RTT of the section of the communication way with the error.
Another problem with high error rates is for example that TCP Reno and other common TCP versions ascribe losses of segments to congestion problems, causing a totally inappropriate activation of congestion avoidance mechanisms [3].

A study has been made as described in [4] which evaluated the TCP flow-rate at different round trip times with very low package loss (1 package loss per 100 million packages) as shown below.

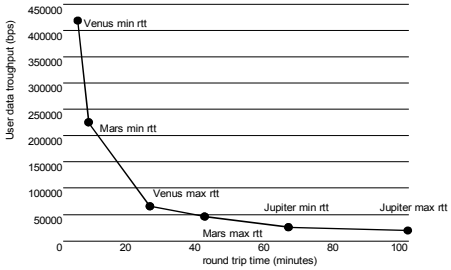



Figure 3: Upper bound on TCP throughput at very low package loss

As there is always some package loss the same experiment has been carried out with a package corruption rate of 1/5000:

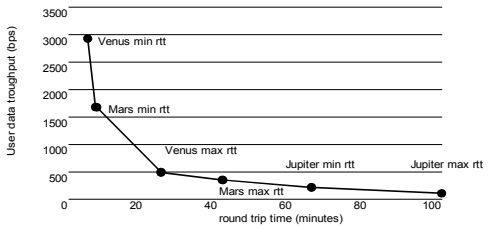



Figure 4: Upper bound on TCP flow-rate at package corruption rate of 1/5000

The most crucial part will be the high signal propagation delay. This could cause an idle timeout because of too late received package acknowledge messages or long delayed additional requests.
TCP includes a startup method called slow-start. The initial very low speed is doubled after each successful package transfer until a maximum value is obtained or the receiver signals that transmission speed is too high. Measurements for the inter planetary special interest group (IPNSIG) showed that this effect is devastating for getting a reasonable transfer speed. The slow-start mechanism already has strong effects at relatively low round trip times (RTT) as seen in figure 5, but at RTT's of 480 seconds with a 1 Mbit/s channel the speed is still below 5 Kbit/s after one hour of operation time.

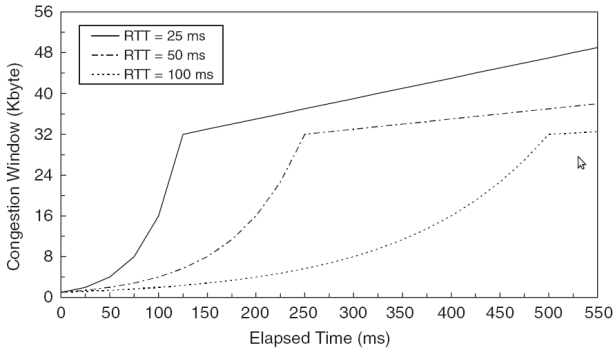



Figure 5: Congestion window evolution in time for several RTT values [3]

Another restriction is the window size limit of 65 Kbytes of TCP in its initial version which limits the bandwidth delay product. The usable bandwidth will be $2^{16}$/(RTT in seconds). For an Earth-Mars maximum round trip time of approximately 1337 seconds the maximum bandwidth would be 49 Bytes/s when not using window scaling.
The sequence numbers are limited too, with a maximum of $2^{32}$ numbers it could happen that too much packets are on the way and the numbers will wrap-around (two packets with the same sequence number) and cause incorrect acknowledges. The maximum segment lifetime (MSL) for an error free operation can be calculated according to [5] by $2^{31}$/bandwidth in bytes. The following table shows the maximum segment lifetime at different bandwidths and should indicate that large data rates and large delays exclude each other.

| Network | Bandwidth in bit/s | Time until segment number wrap in s |
|---|---|---|
| Modem 56K | 56 Kbit/s | $3*10^5$ (~3.6 days) |
| Double ISDN | 128 Kbit/s | $1.3*10^5$ (~1.5 days) |
| DSL | 1.5 Mbit/s | $10^4$ (~3.2 hours) |
| FDDI | 100 Mbit/s | 170 |
| Gigabit | 1 Gbit/s | 17 |

Table 1: TCP maximum segment lifetime at different bandwidths

In recent years several solutions have been proposed to improve TCP performance. This includes SACK (Selective ACK) to quickly recover from losses or TCP Peach which replaces slow start and fast recovery by sudden start and rapid recovery. Or TCP Extensions for High Performance that increase the bandwidth

delay product by scaling the window size and considering the packet timestamps at packages with equal sequence number.

In communication conditions that share delays with satellite links TCP Hybla which is analyzed in [3] seems to perform best. TCP Hybla assembles a set of enhancements. It modifies among others the congestion window increase to something called "constant rate additive increase" and will use SACK's and timestamps.

The problem that TCP requires an end to end connection still remains, but this could be avoided by splitting the connection into a hop by hop communication at upper layers. The whole path will then consist of devices establishing TCP connections to their neighbors but not from end to end.

### *3.1.3 UDP And Others*

The User Datagram Protocol does not suffer from the problem of many exchanged messages due to hand shaking until transfer start or a limited number of packets are on the way. The design of UDP as an unreliable protocol inhibits an exclusive utilization. But with the usage of a convergence layer, UDP can be made reliable. This is done by protocols like Saratoga and Licklider Transmission Protocol (LTP). Saratoga focuses only on efficient communication to the next hop by filling the link with packets sent at line rate [6].

## 3.2 Currently Used Space Communications Systems

From the beginning space missions have needed a wireless connection which can handle long delays.

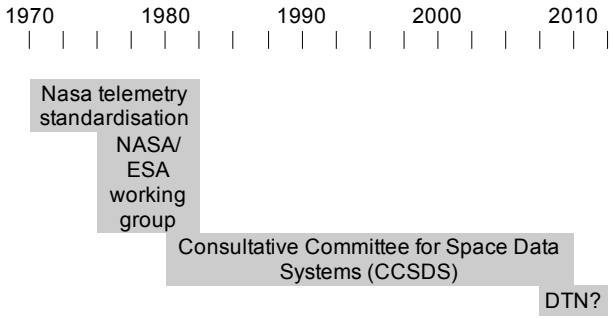



Figure 6: Evolution of space communication

The presentation [7] shows that after the first years of space history it became essential to introduce standardisation to keep control of the complex electronic systems. Later, with the cooperation of different countries the Consultative Committee for Space Data Systems (CCSDS) was formed in 1982. CCSDS is the linking board among most of the existing space agencies for the development of space communication standards. It has worked on various standards including Space Communications Protocol Specifications (SCPS) which mostly extend Internet protocols for the special demands in space.

Newer space missions often include several modules that communicate among each other. This requires a communication network architecture instead of a single link of each module to the Earth, as described in [7]. Recall the example of the moon titan in Section 3.1.2. There is a orbiting sonde, a landing module and probably more units that explore the moon's environment. According to the reference [7] it is expected that standardized space networking architectures and devices allow to build a low cost and reusable infrastructure that can be shared by many diverse space missions.

Among the extensions and adaptation of the Internet protocol by CCSDS an overall solution which is not limited by some backward compatibility or adaption to special missions is missing. Therefore research has been carried out to use DTN for an Interplanetary Internet (IPN).

# 4. DTN

The IETF which is responsible for standardizations concerning the Internet published the latest state of research in the area of DTN in the RFC 4838 [8].

## 4.1 Design Principles

The DTN architecture primarily consists of the bundle layer and some lower protocols. The bundle layer resides on top of a transport layer and is responsible for the end to end communication and uses the lower layers for hop-to-hop communication from DTN device to DTN device.

Lower layer protocols are chosen according to specific needs but they must be capable of supporting the expected delays over one hop (from DTN device to DTN device). The layered architecture of DTN with its overlay network enables the support of heterogenous networks.

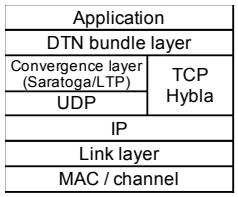



Figure 7: DTN protocol stack

As can be seen in Figure 8, the bundle layer packs applications data units (ADU) and passes them to the Transport protocol [2]. At the lower protocols the communication process works as usual. If the data gets to a DTN capable device, then the data will be passed up to the bundle layer and the transport layer finishes its connection. The data needs to be cached if there is a large delay to the next DTN device. If and when possible the process will continue passing the data down to the transport layer as described in the previous sentence.

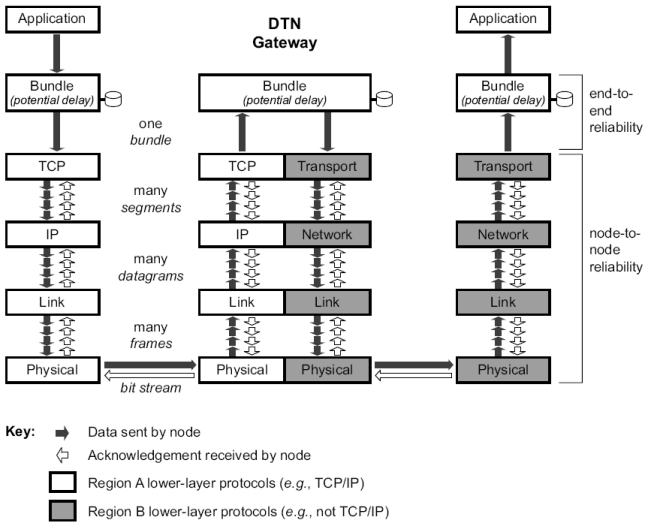



Figure 8: Bundle overlay message transport [2]

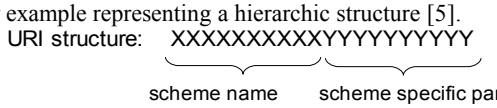
## 4.2 Nodes and Endpoints

A DTN network consists of nodes which are the communication points in the network. If the node is the origin or the final destination of a bundle it is named an endpoint or a host. If a node forwards only bundles in its own DTN region it is called a router or else it is called gateway.

## 4.3 Node Identification

Each node needs to be identified by a unique expression. As each node can act as an endpoint the expression is called endpoint

identifier (EID). This uniform resource identifier has a fixed structure. It starts with a scheme name which will be assigned by the IANA and is followed by a scheme specific part (SSP). The scheme defines how the SSP will be formed, but usually it can be chosen according to the specific tasks of the nodes in the scheme, for example representing a hierarchic structure [5].




The URI can contain any possible character, but non ASCII ones have to be expressed by percent encoded octets. EID's may be used for routing but are not essential.

One specialty of DTN is the usage of late binding, which is a method of Locator/ID split. Which means that the EID to destination mapping will not be made at the senders node like address resolution in the domain name system, but at each hop from node to node. This behavior is advantageous in a system where not all nodes are available all the time or node availability will change during the transmission of data which might happen more often due to the large transfer time.

## 4.4 Connection Types

Except from the common one sender one receiver paradigm, multicast will be supported which maps one destination EID to more endpoints while anycast delivery is successful when one of a group of endpoints receives the transferred bundle. If one node wants to receive bundles of a source it needs to register itself to the specific multicast or anycast group.

DTN nodes might not be persistently connected to the network. The following types are possible:[2]

- Persistent: The node will be available all the time
- On-Demand: The node availability may depend on different events. For example a sensor node when it measures a specific value
- Scheduled: Availability can be predicted, this might be fixed reoccurring interval for example due to a planets rotation or at other specific time intervals
- Opportunistic: This connection type can not be handled very good as the connection time will not be predictable

The knowledge of the connection type is very useful to enhance the delivery process for example by omitting opportunistic contacts in a transmission path. If an opportunistic contact is used it might be useful to send at an redundant path that might be slower but with a predictable transmission time. The knowledge of the scheduled intervals of connectivity are useful for neighboring nodes to optimize the time for broadcasting and to optimize storage.

## 4.5 Data Transfer Method

Store and forward transfer method is used in DTN to support connections to nodes which are not available all the time. Some amount of storage is therefore required at each node, preferably persistent storage to keep the data in the case of system restarts. The storage handling is critical as it is dictated by the available memory but also crucial for successful or time-saving delivery. Bundles will be equipped with a field called useful life indicator which dictates the time of deletion from the nodes storage.

## 4.6 Bundle package, Encapsulation

Applications pass their application data units to the bundle layer. This layer will encapsulate them in bundles whose length will depend on network properties like available storage in the transmission path and others. The bundle header will contain some semantic information necessary for delivery. This includes the originating timestamp, useful life indicator, a class of service designator, length, some delivery options and sender and receiver EID.




Figure 9: Structure of the bundle protocol block [8]

The bundle protocol block is optimized to consume as less overhead as possible while providing high flexibility. This is achieved by Self Delimiting Numeric Values (SDNVs, used at * in the figure above) that encode 7 bit numbers in 8 bit using one bit for number end termination. Further optimized is the storage of the scheme and SSP by storing a reference to the compressed values in the dictionary byte array to avoid redundancy.

## 4.7 Routing and Forwarding

Efficient routing is a challenging task in a DTN. If the connection times of opportunistic or periodic contacts are known in advance or can be estimated routing can be optimized.

The connection of nodes can be visualized by a multi graph whose edges resemble the possible connections. This might be directed and time varying in delay and capacity. The capacity and the interval of connected time multiplied yields to a contacts volume which is a good measurement of the capabilities of the connection and therefore usable for routing decisions if connection time is known ahead of time. In other cases routing calculations are complicated and still under research.

## 4.8 Fragmentation And Reassembly

For an effective utilization of the available volume it is useful to change the size of the bundles to be transferred. This is done by splitting up larger bundles into smaller ones.

- Proactive: If the volume of a connection and the capabilities of the receiving node (which includes available storage, computing capacity depending on other tasks) is known in advance the fragmentation can be evaluated for best delivery
- Reactive: An adaption of the fragmentation during sending process is denoted as reactive. This will enable to use partially received bundles to be forward and as a result will fully use the available volume. The are

additional methods include in DTN called converge layer protocols that inform the sender of the partial transfer. This will then be used to adapt fragmentation at the senders node.

## 4.9 Security

The usage of common authentication and authorization is not always possible. For example methods like Diffie-Hellman key exchange to establish a synchronous communication encryption require many exchanging message and cause a very bad performance at high delays. Access to a central server for privilege inquiry not a good idea as well.

It is useful to follow some rules to avoid flooding of the network by an unauthorized user:

- prevent unauthorized applications at the start to avoid unnecessary transmitted data
- prevent unauthorized application from asserting control over the infrastructure
- control transmission rate and class of service
- discard damaged or improperly modified bundles
- detect and unauthorize compromised nodes as fast as possible

DTN can be used with the Bundle Security Protocol [9] to satisfy the security requirements, that utilizes hop-by-hop and end-to-end authentication and integrity mechanisms.

# 5. Related projects

The KioskNet [10] project is based on the DTN concepts and incorporates the aspects mentioned in Section 2.3 Terrestrial Wireless Networks. It is intended for developing countries to provide Internet access to remote locations with no direct connection. The system functions as follows: A PC is connected at the remote location with a kiosk controller. That device stores the data requests and sent data as long as a mobile computer which might be included in a car comes close to it. Then data will be exchanged wireless by the devices, the mobile device will deliver

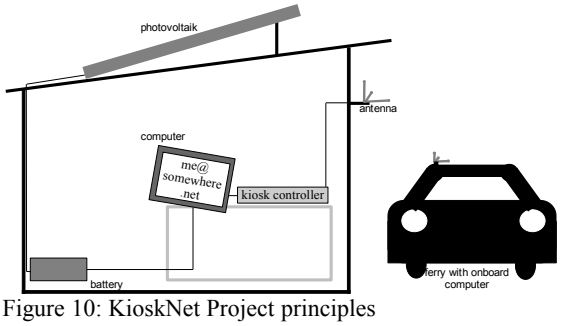


Figure 10: KioskNet Project principles

previously requested data and store data to be sent. Later when the mobile device gets Internet access it will forward the data and collect requested information. A more advanced version can be used when a mobile phone connection is available (this is more likely in developing regions than a wired telephone because of its cheaper installation). Then the priority and type of data will be checked and if necessary transferred via the mobile phone. This might be useful for an urgent message to a doctor but not for common email due to it's higher costs.

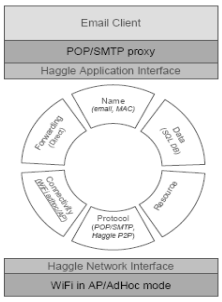


Figure 11: Structure of Haggle architecture

Another example of a DTN is Haggle [11] which enables communication in the presence of intermittent network connectivity. It exploits autonomic opportunistic communication in the absence of end-to-end communication infrastructure. This example while being a DTN application is different from the DTN architecture described and should show the different approaches designing a delay tolerant network. It is completely eliminating layering above the data-link and uses application driven message forwarding between managers instead.

# 6. Current state

DTN is still subject to research. The usage for IPN seems to be most advanced and was successfully tested in October/November 2008 for four weeks. The test involved several relays stations on earth and performed data transfer with the 32.4 million kilometer distant space probe Epoxi which is on the way for comet investigation since 2005. The relay stations have been simulations of Mars lander, orbiter and ground mission operations centers. A DTN implementation of NASA called Deep Space Network was responsible for the data transfers. There will be a more extensive test with improved DTN software in Summer 2009 including a DTN node at the International Space Station (ISS). The full report on this tests can be read at [12].

The further aim at space networks will be an IPN which consists of an interplanetary backbone to connect regional networks like the one on earth with others throughout the solar system.

# 7. Conclusion

A delay tolerant network seems to be a special subject to establish a connection to artificial objects in space, but the examples in Section 2 show that there are several applications that have to deal with high propagation delay or similar challenges like interruption. The use of existing protocols to reduce development effort like TCP comes with some problems that need to be circumvented. The DTN consists mainly of the bundle layer that lays on to of the transport layer. This design makes it possible to support different lower layers and to split the communication path into several segments, which makes DTN more flexible. Apart from the fact that DTN is still under development it has passed some tests and is used in a reduced version already. DTN will be further standardized and has support from several organizations which is a good start to get into widespread usage in this kind of market.

# 8.References

1: Thomas Haenselmann, Sensornetworks, 2006, http://www.informatik.uni-mannheim.de/~haensel/sn_book/sensor_networks.pdf

2: Forrest Warthman, Delay Tolerant networks (DTNs) a Tutorial, 2003, http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf

3: Carlo Cainin, Rosario Firrincieli, TCP Hybla: a TCP enhancement for heterogeneous networks, 2004, http://www.cs.utk.edu/~dunigan/ipp05/hybla.pdf

4: Robert C. Durst, Patrick D. Feighery, Keith L. Scott, Why not use the Standard Internet Suite for the Interplanetary Internet, 1998, http://www.ipnsig.org/reports/TCP_IP.pdf

5: V. Jacobson, R. Braden, D. Borman, TCP Extensions for High Performance (RFC 1323), 1992, http://www.ietf.org/rfc/rfc1323.txt

6: Lloyd Wood, Wesley M. Eddy, Will Ivancic, Jim McKim, Chris Jackson, Saratoga: a Delay-Tolerant

Networkingconvergence layer with efficient link utilization, 2007, http://personal.ee.surrey.ac.uk/Personal/L.Wood/publications/internet-drafts/draft-wood-tsvwg-saratoga/wood-eddy-saratoga-dtn-iwssc-07.pdf

7: Scott Burleigh, Vint Cerf, Robert Durst, Kevin Fall, Adrian Hooke, Keith Scott, Leigh Torgerson, Howie Weiss, Interplanetary Internet Presentation, 2003

8: V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss, Delay-Tolerant Networking Architecture, 2007, http://www.ietf.org/rfc/rfc4838.txt

9: S.F. Symington, S. Farrell, H. Weiss, P. Lovell, Bundle Security Protocol Specification Draft, 2009, http://tools.ietf.org/html/draft-irtf-dtnrg-bundle-security-08

10: University of Waterloo, The KioskNet Project, 15.06.2009, http://blizzard.cs.uwaterloo.ca/tetherless/index.php/KioskNet

11: Christophe Diot, Haggle: Situated and Autonomic Communications, 13.07.2009, http://www.haggleproject.org/index.php/Main_Page

12: Dwayne Brown, Katherine Trinidad, Successfully Tests First Deep Space Internet, 18.11.2008, http://solarsystem.nasa.gov/deepimpact/press/news-display.cfm?News_ID=30315

# Identity Management

Johannes Schlicker
Betreuer: Holger Kinkelin
Seminar Innovative Internettechnologien und Mobilkommunikation SS 2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: schlicke@in.tum.de

## KURZFASSUNG

Elektronische Identitäten werden verwendet, um sich als Person im Internet zu identifizieren. Identity Management ist eine häufig eingesetzte Technologie, solche elektronischen Identitäten verschiedenster Services zu vereinigen. Das Hauptziel der Anwendung ist einerseits, dem Benutzer einfacheren Zugriff auf verschiedene Ressourcen zu ermöglichen und andererseits bieten sich auch dem Service, der Identity Management einsetzt viele neue Möglichkeiten, wie beispielsweise, die Echtheit eines Benutzer zu überprüfen. In dieser Ausarbeitung werden aus Benutzer- und Anbietersicht verschiedene Punkte diskutiert.

## SCHLÜSSELWORTE

Benutzerverwaltung, IDM, Identity Management System, Single Sign-On, Internet, OpenID, Phishing, Shibboleth, Windows Cards pace

## 1. EINLEITUNG

Das Internet, wie es heute existiert ermöglicht einem Nutzer vielseitige Verwendung. Dies erfordert meistens eine Registrierung bei Services die man nutzen will. Eine Registrierung ist gleichzusetzen mit der Erstellung einer elektronischen Identität, als welcher man sich bei einem bestimmten Service ausgibt. Die Einsatzbreite der weltweiten Vernetzung ist immens und bedarf einer großen Anzahl solcher Identitäten, wenn man die Möglichkeiten, die der Einsatz von Internettechnologien bietet ausschöpfen will. Vor allem für Normalanwender, die das Internet täglich sowohl für den Einsatz am Arbeitsplatz, als auch privat sehr häufig nutzen, besteht die Notwendigkeit, sich bei vielen Diensten anzumelden. So besitzt ein Nutzer beispielsweise bei *GMail* E-Mail-Konto, ist bei *Facebook* registriert und führt seine Telfongespräche über *Skype*.

**Abbildung 1: Identitätsverwaltung ohne IDM**

Als Folge dieser Registrierungen gibt es nun eine gewisse Anzahl von Accounts (siehe Abbildung 1), die jeder Benutzer privat verwalten muss. Diese Verwaltung beinhaltet das ständige Aktualisieren der Benutzerdaten, die Verwaltung vieler Kombinationen aus mindestens Benutzername und Passwort und vieles mehr. Um den Zeitaufwand dafür möglichst gering zu halten und um weitere Vorteile zu genießen, besteht nun die Möglichkeit einer zentralen Identitätsverwaltung. Allerdings birgt ein solches System nicht nur Vorteile, sondern verschärft auch vorhandene Sicherheitsrisiken und schafft unter Umständen sogar neue Probleme.

Die Lösung für die zentrale Verwaltung digitaler Identitäten bieten sogenannte Identity Management Systeme in verschiedenen Ausprägungen. In diesem Artikel wird primär auf den Einsatz von Identity Management im Internet eingegangen. Der technische Aufbau, der Nutzen für den Endbenutzer und die Anforderungen, die es erfüllen muss stehen dabei als Hauptpunkte im Vordergrund.

Das erste Kapitel wird einen Einblick in IDM geben. Dies umfasst auch die allgemeine Funktionsweise. Im Anschluss daran werden Vor- und Nachteile diskutiert und an verschiedenen Beispielen heute aktuelle Ansätze erläutert.

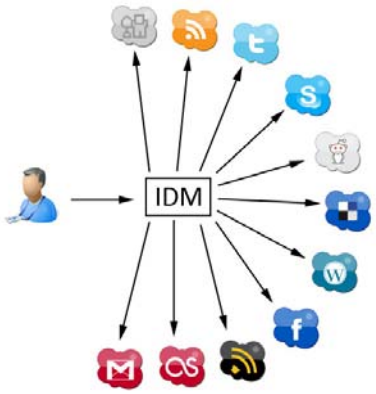



**Abbildung 2: Identitätsverwaltung mit einem IDM System**

## 2. EINFÜHRUNG IN IDENTITY MANAGEMENT

### 2.1 Terminologie

Um im weiteren Verlauf des Dokuments Sachverhalte anschaulicher erklären zu können, sind im Folgenden die wichtigsten Begriffe kurz erklärt:

- IDM: Abkürung für *Identity Management System*
- Consumer: Service im Internet, der einen IDM-Dienst nutzt

- Identity Provider: Speicherort von Identitäten, die von einem Nutzer angelegt werden können
- Phishing: Vortäuschen einer Login Seite, um illegal Zugangsdaten einer anderen Person zu erhalten.

## 2.2 Aufgaben und Funktionen eines Identity Management Systems

Ein Identity Management System hat die Aufgabe verschiedene Aufgaben und Funktionen zu erfüllen, die es dem Benutzer vereinfachen sollen, sich im Internet aufzuhalten.

### *2.2.1 Verwaltung digitaler Identitäten*

Eine zentrale Aufgabe und zugleich die Grundidee hinter diesem System ist die Verwaltung mehrer Identitäten. Dabei ist zu beachten, dass eine Identität eindeutig auf eine Person rückführbar ist, aber eine Person hinter vielen verschiedenen Identitäten im Internet auftreten kann. Die Notwendigkeit mehrere Identitäten annehmen zu können, besteht darin, dass man anderen im Internet agierenden Personen nicht allen die gleichen Informationen zu Verfügung stellen will. So sollten in einem Forum keine Informationen erscheinen, die man für eBanking hinterlegt hat. Um dies zu vermeiden erstellt man eben genau eine Identität für das Forum und eine Identität für eBanking, die man zu entsprechenden Zwecken einfach annehmen kann.

### *2.2.2 Single Sign-On*

Ein weiterer sehr wichtiger Bestandteil und daher in der Regel in allen Identity Management Systemen implementiert ist Single Sign-On (SSO). SSO bezeichnet ein Verfahren, bei dem die einmalige Anmeldung beim Identity Service Provider ausreicht, alle weiteren Dienste die SSO unterstützen ohne weitere Anmeldung nutzen zu können. Ein zweiter wichtiger Punkt ist, dass nicht nur Authentifizierungsinformationen einmalig vorhanden sind, sondern dass diese auch nur ein einziges Mal eingegeben werden müssen, um internetübergreifend auf alle Dienste den Zugriff zu erhalten, für den man berechtigt ist.

### *2.2.3 Bereitstellung von Informationen*

Um einen weiteren Schritt Richtung „einfaches Internet“ zu machen bieten Identity Management Systeme die Möglichkeit verschiedene Informationen zu einem Nutzer zu speichern. So können personenbezogene Angaben, wie Name, Geschlecht oder die eMail Adresse gemacht werden, sowie Informationen, die nur einzelne Identitäten betreffen, wie einen Nickname für eine Identität, die man in einem Forum annimmt, oder Kontonummer und Bankleitzahl für einen Electronic Banking Account. Es können hier also beliebige Daten angegeben werden. Eine Anforderung hierbei ist, die Informationen natürlich nicht nur zu speichern, sondern auch bei Bedarf zur Verfügung zu stellen. Dies wird immer durch einen Identifikator geregelt. Dieser leitet auf eine die Authentizität prüfende Seite weiter und gibt die angeforderten Informationen nach erfolgreicher Authentisierung frei.

## 2.3 Funktionsweise

Im Folgenden wird die Funktion eines Identity Management Systems erklärt. Hier ist außerdem zu erwähnen, dass es für den Consumer grundlegend zwei Möglichkeiten gibt eine Identity Management Service zu nutzen. Einerseite kann er gänzlich auf eine eigene Benutzerverwaltung verzichten und diese komplett dem Service Provider überlassen, oder er speichert Benutzerdaten ganz oder teilweise auf eigenen Servern und verwendet den Service Provider nur als dritte Instanz, die nur für die Benutzerauthentifizierung zuständig ist.

Die beiden folgenden Ansichten erklären, was der Nutzer an Vereinfachung erreichen kann und was allgemein hinter den Kulissen passiert.

### *2.3.1 Aus Sicht des Benutzers*

Dem Benutzer bieten sich natürlich viele Möglichkeiten ein Identity Management System zu nutzen. Ein Beispiel, das typischen Ablauf aufzeigt, ist die Authentifizierung gegenüber einem Service im Internet mittels Identity Management.

Der Benutzer will bei einem Dienst im Internet einloggen. Dazu besucht er die Startseite dieses Dienstes und wird recht schnell ein Textfeld finden, in welches er nur seinen vom Service Provider zur Verfügung gestellten Identifikator eingeben muss. Dieser Identifikator ist heute in den meisten Fällen eine URL, die eine Person eindeutig identifiziert. Über diesen Identifikator leitet der Browser dann auf die Login Seite des Service Providers weiter. Sollte der Login zu einem früheren Zeitpunkt noch nicht stattgefunden haben wird dieser mit dem einzigen Benutzernamen und Passwort, das der Benutzer noch besitzt durchgeführt. Nach korrekter Eingabe der Login Daten, kann der Benutzer eine Identität auswählen, oder die für diesen Dienst voreingestellte verwenden. Dann leitet der Browser zurück auf die Seite des Dienstes und teilt diesem mit, dass der Login erfolgreich war und alle Informationen die unter der ausgewählten Identität gespeichert wurden stehen nun zur Verfügung. Außerdem wird dem Benutzer nun der ihm zugestandene Zugriff gewährt und er kann somit den Dienst nutzen.

### *2.3.2 Aus technischer Sicht*

Da sich einzelne Anbieter teilweise in ihrer technischen Ausarbeitung unterscheiden, gehe ich nur allgemein auf diese Sichtweise ein.

Es gibt so beispielsweise verschiedene Ansätze eine Benutzerauthentifizierung durchzuführen. Diese kann über eine Benutzername/Passwort Kombination überprüft werden, aber es sind auch andere Methoden, wie Stimmerkennung oder Fingerabdruckprüfung denkbar. Somit kann hier allgemein nur von Benutzerauthentifizierung gesprochen werden.

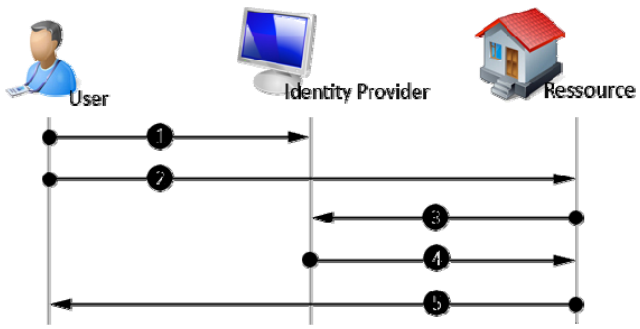



**Abbildung 3: Technischer Ablauf einer Authentifizierung (verinfacht)**

Im ersten Schritt authentifiziert sich der Benutzer gegenüber dem Identity Provider. Dies geschieht, wie angesprochen über Benutzername und Passwort, oder andere Möglichkeiten. Der Identity Provider prüft diese Daten nun mit den gespeicherten ab und gewährt jeder einzelnen Identität der Person, die sich eingeloggt hat Zugang zu entsprechenden Services, für die sie hinterlegt wurde. Außerdem werden lokal auf dem Anwendercomputer weitere Informationen häufig als Cookie gespeichert. Wichtig ist dabei die Gültigkeit des Logins (beispielsweise 24 Stunden).

Wenn in Schritt zwei der Benutzer nun Zugriff auf eine Ressource im Internet erfragt, so muss diese vorher beim Identity Provider

überprüfen, ob der anfragende Benutzer mit der entsprechenden Identität überhaupt Zugriff hat und wenn ja, ob der Benutzer sich bereits authentifiziert hat.

Dazu schickt der Consumer im dritten Schritt eine Anfrage an den Identity Provider. Wer der Identity Provider ist, wird durch den Identifikator festgelegt. Dieser hat also nicht nur die bereits erwähnte Aufgabe, eine Person zu identifizieren, sondern ist auch eindeutig auf einen Service Provider rückführbar. Mit den Informationen *wer* sich bei *welchem Dienst* mit *welcher Identität*, die bei *welchem Identity Provider* hinterlegt ist, kann der Identifikator nun richtig weiterleiten und es wird überprüft, ob eine zugriffsberechtigte Identität vorhanden ist und ob die zugehörige Person eingeloggt ist.

Da in Schritt eins der Login bereits stattgefunden hat wird die Zugriffsberechtigung in Schritt vier erteilt und schließlich im letzten Schritt die Ressource an den Benutzer freigegeben.

Sollte sich der Benutzer nun beim Service Provider ausloggen, oder die Gültigkeit des Logins abgelaufen sein, wird er natürlich global mit allen Identitäten ausgeloggt und kann auf die Dienste erst nach erneutem Einloggen zugreifen.

## 2.4 Anforderungen an ein IDM System

Um Vorteile hervorzuheben und um Nachteilen möglichst stark entgegentreten zu können, wird eine Liste von Anforderungen gestellt. Diese können individuellen Einsatzgebieten angepasst sein, aber viele dieser Anforderungen sind allgemeingültig und sollten bei der Implementierung jeder Identity Management Lösung beachtet werden.

Die im Folgenden genannten Anforderungen sind heute bereits teilweise implementiert. Für die nicht beachteten Anforderungen sind Lösungen entweder schwer zu entwickeln, oder auch in der heutigen Internetumgebung schlicht nicht durchführbar. Deshalb werde ich Anforderungen auflisten und teilweise auf Probleme eingehen, aber keine Lösungsansätze vorschlagen.

### *2.4.1 Korrektheit gespeicherter Informationen*

Viele Anbieter sind auf die Korrektheit und Aktualität identitäts- und personenbezogener Informationen angewiesen. Außerdem ist es äußerst essentiell, dass diese Daten zu jedem Zeitpunkt verfügbar und somit für einen Consumer abrufbar sind.

Beispielsweise kann ein Onlineversand für Bücher sein Geschäft nicht betreiben, sollten Angaben wie die Anschrift, oder auch Bankdaten nicht mit hoher Sicherheit der korrekt, oder erst gar nicht vorhanden sein. [5]

### *2.4.2 Erweiterbarkeit*

Man kann vor Allem im Internet heute nicht sagen, was morgen sein wird. Ein Identity Management System, das bei vollständiger Integration eine sehr zentrale Rolle spielt muss somit möglichen Änderungen des Internets oder sogar neuen Anforderungen standhalten können. Dabei ist nicht nur an das einfache Erweitern des Systems gedacht, wobei beispielsweise neue Eingabefelder, oder sicherere Authentifizierungsoptionen Hauptpunkte wären, sondern auch an tiefe technische Umstrukturierungen.

In naher Zukunft muss beispielsweise der Adressbereich von IPv4 auf IPv6 umgestellt werden. In manchen Systemen sind IP basierte Sicherheitsmechanismen eingebaut und wurden Jahre auf Sicherheitslücken getestet. Sollte der Umstellung nun vorgenommen werden, muss ausgeschlossen, dass dadurch die Sicherheit oder allgemein die Funktionalität eingeschränkt wird. [8]

### *2.4.3 Verhaltensanonymität*

Die meisten Identity Management Systeme, die bereits heute auf dem Markt biete die Möglichkeit zu überprüfen, was man in den letzten Tagen, Monaten oder Jahren gemacht. Der Benutzer hat die Möglichkeit in einem Logbuch nachzuverfolgen, für welche Dienste er sich eingeloggt hat. Ein Logbuch wird hier als Protokoll verwendet, welches abspeichert, wann der Benutzer welchen Dienst genutzt hat. Es ist allerdings fragwürdig, ob das Speichern dieser Daten wirklich im Sinne des Benutzers sein kann. Denn sind Daten einmal gespeichert hat unter Umständen nicht nur der Benutzer darauf Zugriff, sondern auch jeder Datenbankadministrator, oder im schlimmsten Falle sogar Dritte, sollte nicht ordnungsgemäß mit den Daten des Benutzers umgegangen werden.

Kann ein Arbeitgeber herausfinden, dass einer seiner Mitarbeiter wöchentlich Schnaps bei einem Onlineversand kauft, darf er diesen zwar offiziell nicht aus diesem Grunde ausstellen, aber wird andere Argumente finden, mit denen er die Beendigung des Arbeitsverhältnisses begründen kann.

Ein anderes Beispiel ist heute, wenn Überwachung und staatlicher Zugriff auf private Daten ein zentrales Thema sind vielleicht von noch höherem Interesse. Sollte sich aus ungenannten Gründen eine Person im Internet Informationen über radikale Terrororganisationen einholen und eine staatliche Instanz erhält unter Umständen sogar legal Zugriff auf das Logbuch dieser Person, könnte dies dazu führen, dass beispielsweise eine stärkere Überwachung von staatlicher Seite angeordnet wird.

Szenarien wie diese machen die Anforderung an Anonymität im Internet sehr wichtig. Windows Cardspace bietet bereits heute ein System an, das diese Anforderung erfüllt.

### *2.4.4 Verifizierbarkeit einer Identität*

Bei den meisten Diensten im Internet ist es möglich sich selbst Accounts zu erstellen, ohne dass überprüft wird, ob die Person die dahinter steht wirklich existiert. Bei manchen Anwendungen kann das zu Problemen führen. Betreiber eines Identity Management Systems müssen also garantieren können, dass sowohl die erstellte Identität, aber auch die reale Person dahinter echt sind. Es ist zu beachten, dass Echtheit nicht nur bedeutet, dass der Account von einem Menschen erstellt wurde und nicht von einem Bot, sondern es muss eine Möglichkeit geben, um sicherzustellen, dass die Person auch wirklich diejenige ist, welche sie vorgibt zu sein.

Es gibt verschiedene Lösungsansätze, die dieses Problem lösen. Im Beispiel Shibboleth werde ich auf einen relativ sicheren Ansatz eingehen. [8]

### *2.4.5 Datensicherheit*

Im Abschnitt 2.3 wurde klar, dass Passwörter, Benutzername und andere sensible Informationen nicht mehr nur zwischen Consumer und Benutzer kommuniziert werden, sondern mit Identity Management diese Informationen auch noch zwischen anderen Identitäten ausgetauscht werden. Diese Übertragungen müssen garantiert sicher sein. Sollte das Abhören der Informationen absichtlich oder unabsichtlich dennoch möglich sein, so wird neben der Problematik des Phishing, auf die ich später noch genauer eingehen werde, eine weitere Sicherheitslücke geschaffen. Natürlich besteht das Problem der sicheren Übertragung bereits heute, aber je wertvoller die abhörbaren Daten sind, desto stärker werden die Anstrengungen diese Daten zu erhalten. Somit muss auch in diesem Bereich noch einiges an Arbeit geleistet werden, damit Übertragungen sicherer werden.

Weiterhin ist zu beachten, dass die beteiligten Entitäten (gedacht ist hierbei Identity Provider, Identity Server und eventuell weitere) diese sensiblen Daten weiterleiten, oder sogar

zwischenspeichern. Auch hier muss es Möglichkeiten für den Benutzer geben, diese Einrichtungen als echt einzustufen. [3]

### *2.4.6 Datenschutz*

Ein bereits in Punkt 2.4.3 angesprochenes Problem ist das Problem der Nachverfolgbarkeit im Internet. Ein ebenso problematisches Thema ist der Datenschutz. Viele Betreiber finanzieren ihre Services ausschließlich durch Werbung, oder beispielsweise den Verkauf von Kontaktdaten der einzelnen registrierten Nutzer. Ein Anbieter eines Identity Management Systems hat Daten gespeichert, die so für dritte äußerst wertvoll sein können, da sie vollständig und mit hoher Sicherheit echt sind.

Man muss daher fordern, dass diese Daten völlig unangetastet auf den Identity Servern verbleiben und niemals ohne die Erlaubnis des Benutzers freigegeben werden dürfen. Hierfür sind auf jeden Fall neue Mechanismen von Nöten, da die offensichtlich ausgereiftesten, die heute bei Mobilfunkanbietern im Einsatz sind, oftmals nicht ausreichen um den Datenschutz zu gewährleisten, wie die Telekom im 2008 erkennen musste. [3]

### *2.4.7 Flexibilität*

Da der Einsatz von Identity Management Systemen mit Single Sign-On viele Möglichkeiten bietet, muss dieses System in allen nur denkbaren Anwendungsfällen reibungslosen Betrieb gewährleisten können.

Eine umfangreiche zentrale Aufgabe ist beispielsweise die Zugriffsrechteverwaltung. Verschiedene Benutzer haben verschiedene Rechte bei verschiedenen Consumern. Da die Organisation bei den meisten Dienstleistern im Internet unterschiedlich ist, muss also die Möglichkeit geboten werden, für jeden Consumer, der einen Identity Management Service nutzt eine angepasste Verwaltung zu erstellen, ohne dabei allzu hohen Aufwand betreiben zu müssen.

### *2.4.8 Benutzerfreundlichkeit*

Als letzter Punkt ist natürlich auch die Benutzerfreundlichkeit eine Anforderung, die eigentlich grundlegend überall laut wird. Vor allem aber bei Identity Management, dessen Hauptziel es ist, dem Endanwender eine einfachere Nutzung des Internets zu bieten, ist diese sehr bedeutend. Aus diesem Grund haben die großen Anbieter heute bei der Entwicklung ihrer Software genau darauf geachtet und sind in den meisten Fällen die Forderung nach Benutzerfreundlichkeit vollkommen nachgekommen.

Dieses Problem löst sich allerdings in den meisten Fällen selbst, da sich schwer zu bedienende Services einfach nicht durchsetzen können und somit nicht benutzt werden. [5]

### *2.4.9 Einfache Integration in bestehende System*

Da einerseits in Identity Management Systemen noch teilweise große Sicherheitslücken vorhanden sind und andererseits die Internetnutzer selbst von den Vorteilen eines solchen Systems noch nicht gänzlich überzeugt sind, kommt bei den meisten Dienstanbietern im Internet eine Integration noch nicht in Frage. Sollten diese Probleme und Unklarheiten auf der technischen Seite, aber auch auf Seiten der Consumer so minimiert werden, dass ein Einsatz im täglichen Gebrauch rentabel und sinnvoll wird, dann muss garantiert sein, dass die Integration eines Identity Providers, oder sogar die vollständige Auslagerung der Benutzerdatenverwaltung einfach und schnell verläuft, damit der Consumer nicht durch schwere Administrationsarbeit, oder sogar hohe Umstellungskosten abgeschreckt wird.

# 3. VORTEILE

Sowohl für den Benutzer, als auch für den Endanwender ergeben sich aus der Benutzung von Identity Management verschiedene Vorteile.

## 3.1 Für den Benutzer

Die Vorteile für den Benutzer eines Identity Management System sind teilweise sehr offensichtlich, aber da vielen Menschen IDM im Allgemeinen gar keine Begriff ist, werde ich hier noch genauer darauf eingehen.

### *3.1.1 Ein Login für alles*

Es hat sich in den letzten Jahren immer mehr herauskristallisiert, dass Anwender das Internet in immer mehr Bereichen des täglichen Lebens einsetzen. Dadurch entstand die Notwendigkeit für viele Anbieter, eine Benutzerverwaltung einzusetzen, um personenbezogene Daten abspeichern und zuordnen zu können. Für den Nutzer bedeutet das, dass er sich bei vielen Diensten anmeldet und für jeden dieser Dienste eigene Zugangsdaten hat. Mit Identity Management werden diese einzelnen Zugänge durch einen einzigen globalen ersetzt. Die dadurch erreichte Zeitersparnis kann also nur im Sinne des Anwenders sein.

Ein weiteres Problem, das durch einen einzigen Login gelöst wird, ist, dass man sich nur noch einmalig Zugangsdaten merken muss. Diese müssen nicht natürlich nicht immer aus einer Kombination von Benutzername und Passwort bestehen, aber in jedem Fall ist es ein Fortschritt, sich global anmelden zu können natürlich unter Berücksichtigung von Sicherheitskriterien.

### *3.1.2 Zentrale Bereitstellung von Informationen*

Wie bereits erwähnt, benötigen viele Services eigene Benutzerverwaltungen und können diese nicht komplett auslagern, aber auch in diesem Fall hilft ein Identity Management System. Es besteht für den Benutzer nicht mehr die Notwendigkeit, alle benötigten Daten anzugeben, die ein Dienst benötigt. Die Angabe eines Identifikators reicht aus, um alle Informationen zur Verfügung zu stellen, die benötigt werden.

Beispielsweise muss man sich in einem Onlinelexikon auf Grund der großen Informationsmengen, die pro Benutzer anfallen können eigens registrieren, wenn auch die Benutzerauthentifizierung extern abgewickelt wird. Die Registrierungsmaske verlangt Benutzername, Passwort, eMail Adresse und eventuell weitere Angaben. Diese werden durch die Angabe des genannten Identifikators, der heute oftmals durch eine URL dargestellt wird, automatisch eingetragen und der Benutzer muss schließlich nur noch bestätigen. Sogar die Überprüfung der Daten ist hinfällig, da dies bereits bei der Erstellung der entsprechenden Identität geschehen ist.

### *3.1.3 Zentrale Datenspeicherung*

Neben den Problemen, die die zentrale Datenspeicherung aufwirft, birgt diese natürlich auch immense Vorteile für den Benutzer. Aus Sicht der Datensicherheit und des Datenschutzes weiß der Benutzer genau, wo seine Daten gespeichert sind und kann somit einfacher überprüfen, dass die vertrauliche Behandlung der Daten gewährleistet wird.

Ein weiterer Vorteil betrifft die Datenpflege. Sollte ein Benutzer seinen Wohnsitz ändern, reicht es aus die Daten einmalig zu ändern und alle Services im Internet wissen somit automatisch von der Änderung der Heimatadresse. Ein Umzug ohne Identity Management System würde bedeuten, dass man zum Beispiel seine Daten bei Yahoo, eBay, Amazon und eventuell vielen weiteren Diensten, die Produkte liefern ändern müsste und oftmals weiß man selbst nicht genau, wo man welche Daten hinterlegt hat.

Dies führt auch auf einen nächsten Vorteil. Der Benutzer hat immer den Überblick, welche Daten welchem Drittdienst zur Verfügung stehen und kann diese durch zentrales Ändern von Identitätsinformationen kontrollieren, ohne sich dabei auf verschiedensten Seiten einloggen zu müssen. Dies betrifft natürlich auch das Abmelden von Diensten, was oftmals sehr aufwändig sein kann.

## 3.2 Für den Consumer

### *3.2.1 Datenverwaltung*

Der Consumer von Identitäten, die von einem IDM Dienst angeboten werden, hat zwar unter Umständen nicht mehr eine so hohe Kontrolle über die einzelnen Benutzer, die bei diesem angemeldet sind, aber kauft sich dafür von jeglicher Benutzerdatenverwaltung frei. Er kann auch davon ausgehen, dass die Informationen aktueller gehalten werden, als das ohne Identity Management der Fall war. Hier wird dem Fall vorgebeugt, dass Internetnutzer entweder ignorieren, dass Daten aktuell zu halten sind und diese nur bei Diensten pflegen, die sie oft nutzen, oder aber absichtlich falsche oder unvollständige Daten angeben, um sich auf irgendeine Art zu schützen.

Natürlich bedeutet das auch, dass Kapazitäten eingespart werden können. Dies betrifft einerseits Administrationsaufwand und andererseits Serverlast. Beides wird bei einer Auslagerung der Datenverwaltung logischerweise reduziert.

### *3.2.2 Benutzerauthentizität*

Einem Consumer ist es natürlich wichtig, dass seine angemeldet User seinen Dienst auf ehrliche weise nutzen. Allerdings bieten sich für Anwender des Dienstes teilweise große Vorteile, mehrere Accounts anzulegen, oder sogar automatisch anlegen zu lassen. Übergibt man dieses Problem einer höheren Instanz (in diesem Fall also einem IDM Service Provider), deren Aufgabe es auch ist diese Fakeaccounts zu identifizieren, kann man sich der Lösung dieses Problems ebenfalls entledigen. Fakeaccounts sind Benutzerkonten, die von nicht authentischen Personen angelegt wurden. Außerdem stehen einem Identity Management Provider bessere Werkzeuge zur Verfügung, einerseits einem Erstellen eines Fakeaccounts entgegenzuwirken und sollten doch welche registriert werden, können diese einfacher erkannt und gebannt werden.

Also Beispiel dafür ist der Bookmarkingdienst *Ma.gnolia* zu nennen, der es Benutzern ermöglicht Seiten positiv oder negativ zu bewerten. Stark positiv bewertete Seiten stehen dann in einer Liste weiter oben und werden unter Umständen öfter besucht. Um eigen Seiten so in diese Liste weiter nach oben zu bringen, erstellten viele Benutzer Fakeaccounts, mit denen sie für ihre eigenen Seiten positive Stimmen abgaben. Da 75% aller bei registrierten Accounts nicht echt waren, hat Ma.gnolia beschlossen, die Benutzerdatenverwaltung komplett OpenID zu überlassen. Somit wurde dem erstellen von falschen Konten eine stärkerer Riegel vorgeschoben.

# 4. NACHTEILE

### *4.1.1 Allgemein*

Es ist hier anzumerken, dass es ausschließlich für Consumer eigentlich nur minimale Nachteile gibt. Ein einziger erwähnenswerter dieser Nachteile ist die geringere Kontrolle über Benutzerdaten, der allerdings dadurch kompensiert werden kann, dass bestimmte Daten zu Benutzern auf eigenen Servern gespeichert werden.

Ein Nachteil der sowohl den Benutzer, als auch den Dienst betrifft ist die hohe Tragweite von Systemausfällen. Sollte das System eines Identity Management Providers ausfallen, sind auch die angebotenen Dienste vieler Consumer nicht mehr erreichbar, was in manchen Fällen gravierende Folgen haben kann. Ein Beispiel wäre das Angebot eines Terminplaners. Wenn wichtige Termine nicht eingesehen werden können, obwohl die Daten zur Verfügung stehen würden und auch die Möglichkeit nicht besteht per eMail andere Personen zu fragen, wann ein bestimmter Termin ist, weil ein einziger Server ausgefallen ist, der den zentralen Login verwaltet, dann ist einerseits der Endanwender hilflos und andererseits ist es natürlich auch nicht im Sinne des Consumers, wenn sein angebotener Service nicht zur Verfügung steht.

### *4.1.2 Für den Benutzer*

Gravierende Nachteile bestehen eigentlich nur für den Endanwender und betreffen meist die Datensicherheit und den Datenschutz.

Das immer aktuelle Thema Phishing wird natürlich wieder interessanter für das Verbrechen im Internet. Sollte ein Benutzer auf eine solche Attacke hereinfallen, gewährt er damit einem unautorisierten Dritten nicht mehr nur Zugriff auf einen einzigen Service, sondern auf alle Dienste die der Anwender nutzt. Darunter könnten in Zukunft auch sehr sensible Anwendungen, wie eBanking oder eMail fallen.

Der Vollständigkeit halber sollte hier auch das angesprochene Problem des Protokolls erwähnt werden. Ein solches Logbuch kann natürlich schwere Probleme verursachen, sollte Personen Einblick gewährt werden, die vom Benutzer nicht dazu autorisiert wurden.

Identity Management allgemein verlangt natürlich auch von Benutzern, die sich gänzlich dagegen aussprechen, dass sie sich darauf einlassen, wenn sie bestimmte Dienste nutzen möchten. Wenn sich IDM also durchsetzt und eine Person weigert sich aus genannten nachteiligen Gründen diesen Service zu nutzen, wird diese aus großen Teilen des Internets ausgeschlossen.

Ein weiteres Problem ist, dass es bei neu aufkommenden Änderungen natürlich sofort viele Personen gibt, die die neuen Möglichkeiten erkennen und als Anbieter neuer Dienste agieren möchten. Consumer können sich aussuchen, welchen Dienst sie für Identity Management nutzen möchten und als Folge kann es sein, dass man sich als Endnutzer wiederum bei mehreren Service Providern anmelden muss, um alle Dienste nutzen zu können, die man nutzen will. Diese Tatsache hat zur Folge, dass viele Vorteile (ein einziger Login für alles, einfache Datenverwaltung, usw.) zunichte gemacht werden und dem alten System mit vielen Benutzerwaltungen einfach nur eine weitere Instanz zwischengeschaltet wird, die keine weiteren Vorteile mehr bietet.

# 5. DREI IMPLEMENTIERUNGEN

## 5.1 OpenID

Da *OpenID*, die heute am weitesten verbreitete Implementierung eines Identity Management Systems im Internet ist, werde ich genauer darauf eingehen und an diesem Beispiel auch Phishing detaillierter erläutern

### 5.1.1 *Funktionsweise*

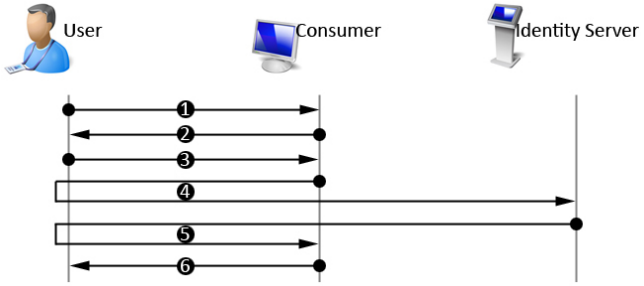



**Abbildung 4: Funktionsweise von OpenID**

Ausgehend davon, dass ein Login bei OpenID bereits stattgefunden hat werden folgende Schritte durchlaufen, bis der geschützte Seiteninhalt einem Benutzer freigegeben wird. Für den Benutzer sind nur die Schritte 1 bis 3 direkt sichtbar nachverfolgbar. Der eigentliche Authentifizierungsprozess in den Schritten 4 bis 6 bleibt hingegen vollkommen verschleiert.

Im ersten Schritt greift der Benutzer auf eine Webseite des Consumers zu. Diese fordert diesen im zweiten Schritt auf seinen Identifikator anzugeben. Im Falle OpenID ist der Identifikator eine URL. Diese hat die Form *http://username.openid.com.* Nach der Eingabe dieser URL in Schritt drei wird als nächstes der Status der Authentifizierung beim Identity Server abgefragt. Da wie anfänglich angenommen der Login bereits stattgefunden hat, wird in Schritt fünf die Bestätigung an den Consumer gesendet und dieser kann schließlich in Schritt sechs geschützte Inhalte an den Benutzer freigeben. Hier ist zu beachten, dass der Browser des Users als zentrale Entität agiert, denn alle Informationen auch nur zwischen Consumer und Provider kommuniziert werden laufen über diesen Browser.

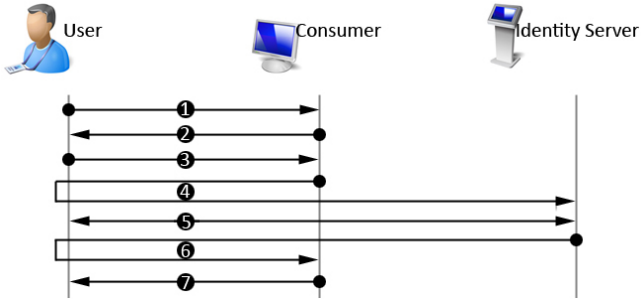



**Abbildung 5: OpenID Funktionsweise mit Authentifizierung**

Unter der anderen Annehme, dass der Benutzer noch nicht eingeloggt ist, kommt ein zusätzlicher Schritt hinzu. So wird jetzt im fünften Schritt durch die Eingabe einer Benutzername/Passwort Kombination die Authentifizierung gegenüber dem Identity Server vollzogen.

### 5.1.2 *Eigenschaften von OpenID*

Eine Eigenschaft, die OpenID beliebt macht bei Consumern ist die leichte Integrierbarkeit. Implementierungen, die auf OpenID basieren wie *myOpenID* sind frei verfügbar. Dies birgt aber auch ein Problem. Jeder einzelne kann als Identity Provider im Internet auftreten. Es ist also nicht garantiert, dass jeder OpenID Anbieter alle Anforderung, die gestellt wurden erfüllt. Primär sind dabei natürlich wieder Datensicherheit und Datenschutz zu nennen. Im Extremfall kann ein solcher Provider natürlich auch direkt gespeicherte Daten nutzen, um Zugriff auf sensible Daten zu erlangen. Beispielsweise könnte ein Arbeitgeber einer größeren Firma einen solchen Dienst anbieten und somit ganz oder teilweise seine Mitarbeiter im Internet nachverfolgen

### 5.1.3 *Phishing*

Als *Phishing* wird eine Attacke bezeichnet, die als einziges Ziel hat, Benutzerdaten von anderen Personen zu erhalten. Mit Identity Management werden, die Daten natürlich wertvoller, da sie Fremdzugriff nicht nur auf eine Ressource gewähren, sondern sofortigen Zugriff auf alle Dienste, für die sich ein Benutzer registriert hat.

Am Beispiel OpenID kann man recht einfach aufzeigen, wie Phishing grundsätzlich funktioniert. Es setzt an Punkt fünf der in Abbildung 5 aufgezeigten Kommunikation an. Sollte ein Benutzer nicht eingeloggt sein, wird er auf die Seite des Service Providers weitergeleitet, wo er sich diesem gegenüber authentifizieren muss. Wenn ein Anbieter allerdings absichtlich nicht auf die Seite des Service Providers weiterleitet, sondern auf eine dritte Seite, die das selbe Aussehen hat, wie die des Service Providers und der Benutzer gibt seine Daten dort ein, ist diesem dritten Anbieter bereits der Zugang gewährt. Oftmals bemerken Internetnutzer gar nicht, dass sie Opfer einer solchen Attacke wurden und vergessen recht schnell, dass der Login nicht funktioniert hat.

Um diesem entgegenzuwirken, sollte man stets darauf achten, dass die URL der Seite auf der man seine Benutzerdaten angibt immer dieselbe ist.

## 5.2 Shibboleth

Als zweite Implementierung, die auch heute bereits Einsatz primär an Hochschulen oder staatlichen Einrichtung findet ist *Shibboleth* zu nennen. Es basiert auf einer Erweiterung von XML namens *SAML*, die Methoden zur sicheren Kommunikation im Internet bereithält.

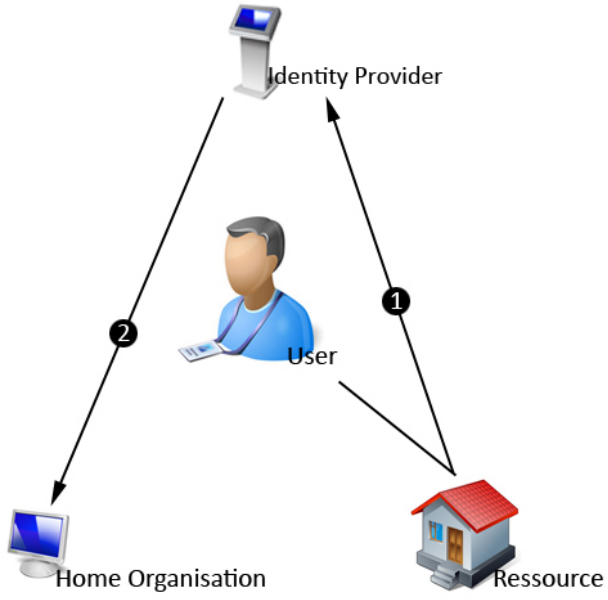



**Abbildung 6: Funktionsweise von Shibboleth**

*SAML* bedeutet *Security Assertion Markup Language*. Die Identitätsverwaltung ist hierarchisch Aufgebaut, was bedeutet, dass es mehrere beglaubigte Organisationen gibt, die Identitäten von verschiedenen Personengruppen bereitstellen. Diese Entitäten können zugleich Consumer sein und Ressourcen für berechtigte Benutzer zur Verfügung stellen.

### 5.2.1 *Funktionsweise*

Der technische Ablauf einer Authentifizierung ist grundlegend dem von OpenID sehr ähnlich. Ein Benutzer der Organisation A

will auf einer Ressource der Organisation B zugreifen. Zweitere überprüft bei einem Service Provider, ob die Berechtigung vorliegt. Der Service Provider sucht im Zuge dessen in Schritt zwei bei der Heimatorganisation A, welche Rechte vorhanden sind und gibt die Ressource dann dem Benutzer frei.

Ein Beispiel dafür ist die Organisation der Münchener Universitäten TUM und LMU, die gegenseitigen Zugriff auf ihre Ressourcen gewähren, wenn sich ein Benutzer als Student einer der beiden Organisationen ausweisen kann.

### *5.2.2 Eigenheiten*

Shibboleth bietet allen beteiligten Consumern einen großen Vorteil. Die einzelnen Benutzer haben keine Möglichkeit eigene Benutzerkonten anzulegen. Der Benutzer kann ausschließlich sekundäre Angaben wie Alter oder Geschlecht ändern. Die hauptsächlich identifizierenden Angaben, wie Name oder Verwaltungsnummern werden ausschließlich von Systemadministratoren erstellt und verwaltet. Das verlangt zwar nach einem stark erhöhten Organisationsaufwand, allerdings ist die Echtheit jedes einzelnen Accounts sichergestellt, was in den Fällen, in denen Shibboleth zum Einsatz kommt meist gerechtfertigt ist.

## 5.3 Windows Cardspace

Eine weitere Möglichkeit IDM zu implementieren verwirklicht Windows Cardspace. Microsoft vergleicht diese Möglichkeit mit einem Geldbeutel. Dieser enthält verschiedene Karten, die je nach Anwendung genutzt werden um sich zu identifizieren.

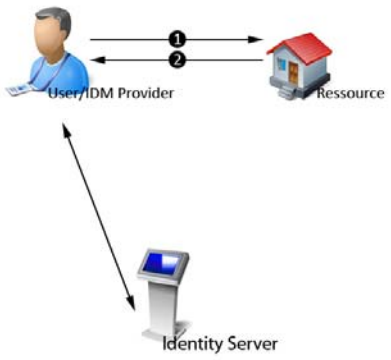



**Abbildung 7: Funktionsweise von Windows Cardspace**

Als primäre Instanzen im Authentifizierungsablauf agieren nur der Benutzer, der zugleich Identity Provider ist und der Consumer. Erfragt ein Anwender Zugriff auf einer Ressource (Schritt eins) fordert diese ihn in Schritt zwei auf sich mit einer Karte zu identifizieren. Es gibt zwei verschiedene Arten von Karten. Einerseits gibt es selbst erstellte Karten, in die der Benutzer alle Informationen einstellen kann, die er will und außerdem gibt es noch verwaltete Karten, die durch einen Drittanbieter, der als Identity Server und zugleich Consumer eintritt erstellt wurden und durch eine Signatur die Echtheit gegenüber diesem gewährleisten soll.

### *5.3.1 Eigenheiten*

Windows Cardspace verhindert, dass der Benutzer im Internet verfolgbar ist und wahrt somit in seiner Grundidee die Anonymität. Allerdings ist nun die Identität des Benutzers nur noch garantiert, wenn sich ein Consumer die Mühe macht verwaltet Karten zu erstellen und einen Service pflegt, der diese Karten verwaltet und administriert.

## 5.4 Vergleich

Die drei Ideen, die hinter *OpenID*, *Shibboleth* und *Windows Cardspace* stehen unterscheiden sich grundlegend voneinander. Es wird hier nur der Hauptunterschied aufgezeigt.

### *5.4.1 Vergleich zwischen OpenID und Shibboleth*

Obwohl der technische Aufbau und der Kommunikationsablauf ähnlich sind, besteht ein Sicherheitsvorteil auf Seiten von Shibboleth. Es kann nämlich garantiert werden, dass alle Benutzer echt sind, was bei OpenID ein nicht komplett lösbares Problem darstellt. OpenID ist *user-centric*, was bedeutet, dass der Benutzer für die Korrektheit der Daten verwantowrtlich ist, wohingegen Shibboleth *institution-centric* ist und somit die Verantwortung auch auf Seiten des Dienstanbieters liegt.

Auch die Gefahr des Phishings ist bei Shibboleth sehr gering, da durch den hierarchischen Aufbau alle teilnehmenden Organisationen bei einer höheren Instanz registriert und akzeptiert sind und unbekannte Mitspieler somit ausgeschlossen werden.

Natürlich ist der Administrationsaufwand bei Shibboleth um eine vielfaches höher als der von OpenID und die Installation, Integration und Erweiterung des Systems gestaltet sich bei Shibboleth ebenfalls sehr viel schwerer.

### *5.4.2 OpenID/Shibboleth – Windows Cardspace*

Windows Cardspace bietet den großen Vorteil, dass der Anwender bei dessen Nutzung keine verfolgbaren Spuren im Internet hinterlässt. Grund dafür ist die ausschließlich lokale Speicherung der Identitäten und somit ist auch nur lokal die globale Einsicht in die Internetnutzung des Anwenders möglich. Im Gegensatz dazu werden das genannte Logbuch bei Shibboleth und OpenID auf Seiten des Service Providers gespeichert.

Der Nachteil von Windows Cardspace ist allerdings, dass nicht in allen Fällen für die Echtheit der Identität und der dahinter stehenden Person garantiert werden kann, mit der sich ein Benutzer im Internet ausgibt.

# 6. EINSATZ UND BEWERTUNG

Identity Management ist heute nur bei wenigen Internetdiensten integriert. Grund dafür könnten die immer noch nicht zur zufriedenstellend geschlossenen Sicherheitslücken sein. Vor allem bei kleineren Anwendungen, wie Webblogs findet OpenID oftmals Einsatz. Der Wikifarm *pbWorks* aber auch größere Firmen wie *Yahoo* haben ebenfalls OpendID vollständig integriert und am Beispiel Ma.gnolia kann man erkennen, dass die Nachfrage auf Consumerseite durchaus besteht. In ihrem Gebiet weltweit in der Spitze agierende Firmen wie Microsoft und GMX unterstützen sogar die Entwicklung von OpenID.

Aus Sicht des Benutzers gibt es nach einer Yahoo Studie noch großen Aufklärungsbedarf. Die Studie ergab, dass den meisten Nutzern Identity Management kein Begriff ist, allerdings sind sie schnell von den Vorzügen der Nutzung überzeugt, wenn man kurz erklärt um was es sich handelt.

Für die Zukunft sollten neuere sichere Authentifizierungsmethoden entwickelt werden, die Techniken wie Smartcards oder Fingerprinterkennung unterstützen, um das Einsatzgebiet zu erweitern. Banken und eMail Anbieter bekunden ihr Interesse, sind aber es Sicherheitsgründen noch zurückhalten mit der endgültigen Integration.

In manchen Kreisen im Internet wird auch von Seiten der Nutzer die Stimme nach Identity Management immer lauter. Die OpenID Gruppe in Facebook wächst stetig und auch in anderen Communities wird das Thema diskutiert.

## 7. LITERATUR


[1] Kim Cameron, Architect of Identity, Microsoft Corporation: The Laws Of Identity (2005)

[2] Rafeeq Ur Rehman: Get Ready For OpenID (2008)

[3] Privacy and Identity Management in Europe: https://www.prime-project.eu/tutorials/gpto/

[4] Single Sign-On Systeme: http://it-republik.de/jaxenter/artikel/Single-Sign-On-Systeme-1499.html

[5] Heise Online. Identity Management: Authentifizierungsdienste mit OpenID. http://www.heise.de/developer/Identity-Management-Authentifizierungsdienste-mit-OpenID--/artikel/136222

[6] OpenID: http://openid.net/what/

[7] Shibboleth: http://shibboleth.internet2.edu/about.html

[8] ifis (Institud für Internetsicherheit: Anforderungen an heutige IdM-Systeme: http://www.internet-sicherheit.de/forschung/aktuelle-projekte/identity-management/anforderungen-an-heutige-idm-systeme/

[9] Michael B. Jones: Introduction To Windows Cardspace: http://research.microsoft.com/en-us/um/people/mbj/papers/CardSpace_One-Pager.pdf (2007)

# Evolution der Kernnetze im Mobilfunk

Martin Veith
Betreuer: Tobias Bandh
Seminar Innovative Internettechnologien und Mobilkommunikation SS 2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: martin.veith@in.tum.de

## Kurzfassung
Mobilfunknetze sind heutzutage allgegenwärtig und müssen immer mehr Anforderungen genügen. Neben der klassischen Telefonie soll auch der schnelle und einfache Zugriff auf das Internet möglich sein. Um diesen steigenden Ansprüchen gerecht zu werden, bedarf es einer stetigen Evolution.

In dieser hier vorliegenden Arbeit werden dazu die verschiedenen Schritte dieser Evolution, angefangen bei GSM und GPRS, über UMTS bis hin zu EPS, im Bereich des Mobilfunkkernnetzes aufgezeigt, und beschrieben welche Vorteile und Auswirkungen die einzelnen Designentscheidungen haben.

## Schlüsselworte
Kernnetz, Evolution, GSM, GPRS, UMTS, LTE, SAE, EPS.

## 1. Einleitung
Mit über 4 Milliarden Nutzern [1] weltweit ist GSM (ursprünglich: Groupe Spéciale Mobile, heute: Global System for Mobile communications) das heute erfolgreichste Mobilfunksystem [2].

Der Grundstein für diese Netzwerkinfrastruktur wurde dabei bereits in den 1980ern gelegt und sollte das analoge Mobiltelekommunikationsnetz revolutionieren und ersetzen. Da diesen analogen Netzen keine länderübergreifende Standards zu Grunde lagen, war es eines der Hauptziele, einen Standard zu entwickeln, welcher in allen europäischen Staaten (und heutzutage fast auf der ganzen Welt) eingesetzt wird [3]. Das Design dieses Standards war dabei vor allem auf die Sprachübertragung ausgerichtet.

Im Laufe der Zeit änderten sich jedoch die Anforderungen. Neben der klassischen Telefonie wird auch die Übertragung von Paketdaten z.B. aus/ins Internet immer wichtiger. Um diese Neuerungen vorantreiben zu können, muss das Mobilfunksystem fortwährend modifiziert und erweitert werden, was zu einer Evolution des Netzes führt. Eine solche Evolution schlägt sich dabei Schritt für Schritt in alle Bereiche des Netzes nieder, sei es auf der Luftschnittstelle und im mobilen Endgerät, als auch im Kernnetz. Diese Veränderungen und Anpassungen im Kernnetz des Mobilfunksystems sind Gegenstand dieser Ausarbeitung.

Dazu werden die verschiedenen Schritte der Evolution, beginnend bei GSM selbst, den Erweiterungen wie GPRS (General Packet Radio Service), der darauf folgende Übergang zum Netz der sogenannten 3. Generation, dem UMTS (Universal Mobile Telecommunications System) und die LTE (Long Term Evolution) bzw. SAE (System Architecture Evolution) als Weiterentwicklungen von UTMS betrachtet.

## 2. GSM
GSM wird als Netz der 2. Generation bezeichnet. Der wichtigste Unterschied zu den Netzen der 1. Generation ist die verwendete Technik. GSM beruht im Gegensatz zu den analogen Netzen vollkommen auf digitaler Technik.

Bei der Entwicklung des GSM Mobilfunknetzes wurden nicht völlig neue Konstrukte verwendet, sondern auf Elemente und Verfahren aus der bereits bestehenden Technik im Festnetztelefonbereich aufgebaut. Dies hat zur Konsequenz, dass im 2G-Netz viele Designentscheidungen aus dem Festnetzbereich übernommen wurden. Von großer Tragweite ist dabei die Entscheidung, dass die (Sprach-)Übertragung innerhalb des Netzwerks leitungsvermittelt erfolgen soll. Zum Aufbau einer Verbindung wird somit eine direkte Verbindung auf einer Leitung zwischen den zwei Parteien geschaltet, welche dann für die Dauer des Gesprächs exklusiv für dieses reserviert ist. Diese leitungsvermittelte Lösung war für die Hauptaufgabe von GSM, der Übertragung von Sprachanrufen, gut geeignet, da dadurch eine konstante garantierte Datenrate und eine fixe Ende-zu-Ende-Verzögerungszeit zugesichert werden konnte. [3]

Die gesamte Architektur bzw. die notwendigen Komponenten zur Erbringung dieses Telefonie-Services werden in den nachfolgenden Absätzen beschrieben.

### 2.1 Architektur
Die Architektur des GSM Netzwerkes lässt sich in drei Subsysteme unterteilen, die im Folgenden näher beschrieben werden:

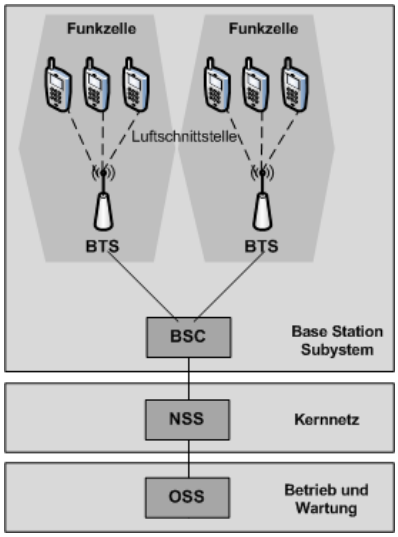



**Abbildung 1 – GSM Architektur**

- Das *Base Station Subsystem* (BSS), auch Funknetzwerk genannt, umfasst alle Komponenten, welche für die Funktionalitäten im Bereich der drahtlosen Kommunikation zwischen mobilen Teilnehmern und der Funkschnittstelle des Netzwerkes benötigt werden und ist in Abbildung 1 skizziert [3].
  Zu diesen Komponenten zählen zum einen die *Base Transceiver Stations* (BTS, Sende-/Empfangsstationen) welche alle funktechnischen Einrichtungen wie Antennen, Signalverarbeitung und Verstärker für die Übertragung beinhalten. Eine solche BTS kann dabei eine (oder auch mehrere) GSM-Zellen mit einem Radius von 100m bis zu 35km aufspannen.
  Als Verwaltungsinstanz für eine Vielzahl solcher Sende- und Empfangsstationen dient eine weitere Komponente des BSS, der *Base Station Controller* (BSC, Feststationssteuerung). Diese Steuerung reserviert Funkfrequenzen, erledigt den Handover zwischen einem BTS zu einem anderen innerhalb eines BSS, führt das Multiplexing der Funkkanäle durch und steuert die Power Control Loop (PCL). Der BSC stellt zudem die Verbindungsbrücke zwischen dem Kernnetz und dem Funknetzwerk dar.

  Das Gegenstück zur Feststation in diesem Funknetz bildet die Mobilstation mit dem eingelegten SIM (Subscriber Identity Module). Durch die Daten (z.B. die International Mobile Subscriber Identity (IMSI)), die auf dieser Smartcard gespeichert sind, kann jeder Teilnehmer im Netz eindeutig identifiziert und authentifiziert werden. [2]

- Das *Network and Switching Subsystem* (NSS), auch Mobilvermittlungssystem genannt [2], stellt das Kernnetz von GSM dar. Es beinhaltet alle Knoten und Funktionalitäten welche für die Verbindung des drahtlosen Netzes mit anderen Netzen (herkömmliches Festnetz oder andere Mobilfunknetze), die Vermittlung von Anrufen, dem Teilnehmer- und Mobilitätsmanagement (Roaming) notwendig sind [3] [2]. Die zugehörigen Komponenten sind in Abbildung 2 zu sehen.

- Das *Operation Subsystem* (OSS), auch Betriebs- und Wartungssystem genannt, erweitert das Netzwerk um weitere Funktionen wie die Authentifizierungszentrale oder das Geräteidentifikationsregister [2].

## 2.2 Kernnetz

Dieser Abschnitt befasst sich nun im Detail mit dem Kernnetz von GSM, welches in Abbildung 2 skizziert ist.

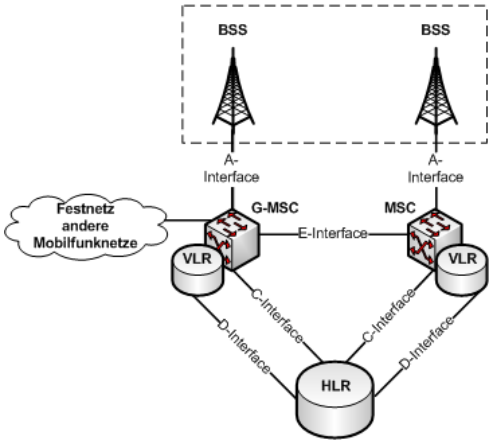



**Abbildung 2 - GSM Kernnetz**

Wie bereits im einleitenden Teil der Architektur beschrieben, dient das Kernnetz hauptsächlich zum Rufaufbau, der Anrufkontrolle und dem Routing der Anrufe von der Quelle zum Ziel und umgekehrt.

Um diese Funktionen zu ermöglichen, gibt es eine Reihe von Komponenten (in Abbildung 2 dargestellt), die in den nachfolgenden Absätzen näher beschrieben werden.

### *2.2.1 Mobile Switching Center (MSC)*

Das MSC, auch Dienstvermittlungsstelle genannt, stellt das zentrale Element des GSM-Kernnetzes dar. Diese hochleistungsfähige Vermittlungsstelle wickelt alle Verbindungen zwischen zwei Teilnehmern ab, indem sie die entsprechende leitungsvermittelte Datenkommunikation herstellt. Diese Kontrollfunktionalität, welche durch das Call Control (CC) Protokoll durchgeführt wird, umfasst folgende Bereiche:

- Registrierung des mobilen Teilnehmers. Sobald ein mobiles Endgerät eingeschaltet wird, muss es sich am Netzwerk registrieren, um sich dort einbuchen zu können.
- Authentifikation des mobilen Teilnehmers. Diese ist im Mobilfunk unbedingt notwendig, da der Teilnehmer nicht wie im Festnetz durch den Zugang zu seinem Anschlusskabel authentisch ist. Die eigentliche Authentisierung wird dabei jedoch nicht vom MSC durchgeführt, sondern nur angefordert.
- Anrufherstellung und Anrufrouting. Das MSC ist für die Zustellung eines Anrufs zuständig. Zur Ausführung dieses Routings ist es notwendig, dass das MSC weiß, wo sich der mobile Teilnehmer befindet und diese Information aktuell gehalten wird. Dazu muss das MSC auch dann sogenannte „Location updates“, also Aktualisierungen der Position durchführen, wenn der Teilnehmer gerade keine aktive Verbindung zum MSC hat.
  Wenn eine solche Positionsänderung während eines aktiven Gesprächs auftritt, so kümmert sich das MSC auch um den dann notwendigen Handover zwischen der alten und der neuen Zelle.
- SMS (Short Message Service)-Weiterleitung. [3]

Die Signalisierung, die für die verschiedenen oben genannten Aufgaben benötigt wird, wird durch das Signalisierungssystem Nr. 7 (SS7) durchgeführt. Durch dieses System werden also das Routing, die Auslieferung von Steuernachrichten, die Einrichtung und Überwachung von Anrufen und die Anrufweiterleitungen ermöglicht. [2]

### *2.2.2 Gateway MSC (G-MSC)*

Eine Gateway MSC hat zusätzlich zu den normalen MSC Funktionen auch noch die Eigenschaft, Verbindungen zu anderen Netzen wie z.B. dem Public Switched Telephone Network (PSTN) weiterleiten zu können [2].

### *2.2.3 Visitor Location Register (VLR)*

Jedem MSC ist ein VLR, auch Besucherregister genannt, zugeordnet. Ein solches VLR ist eine hochdynamische Datenbank [2], welche für jeden Teilnehmer, der gerade vom entsprechenden MSC verwaltet wird, einen Eintrag enthält. Ein solcher Eintrag stellt eine Kopie des originalen Eintrags im Home Location Register (HLR) dar, welches im nächsten Abschnitt noch näher beschrieben wird. Diese Kopien werden angefertigt, um den Signalisierungsaufwand, welcher zwischen MSC und HLR erforderlich ist, zu reduzieren. Eine solche Signalisierung ist z.B. beim Roaming eines Teilnehmers notwendig. Zudem ist es oft so,

dass diese zwei Komponenten unter Umständen geographisch weit voneinander entfernt sein können. Das VLR hingegen ist ja unmittelbar mit dem MSC verbunden, wodurch die Schreib- und Lesezugriffe sehr schnell durchgeführt werden können. Verlässt ein Teilnehmer den abgedeckten Bereich eines MSCs und tritt in den eines anderen MSCs ein, so fordert der neue MSCs wiederum eine Kopie aus dem HLR an, und der Eintrag im alten MSC wird gelöscht. [3] [2]

### *2.2.4 Home Location Register (HLR)*

Das Home Location Register stellt die wichtigste Datenbank im GSM-Netz dar [2]. Sie enthält für jeden beim entsprechenden Mobilfunkprovider registrierten Kunden einen Eintrag, welcher folgende Informationen enthält:

- (logische) Rufnummer (Mobile Subscriber ISDN Number, MSISDN)
- IMSI
- die dem Kunden zugänglichen Dienste z.B. Anrufweiterleitung, GPRS
- Authentifizierungsdaten
- sich dynamisch ändernde Daten wie der aktuelle Aufenthaltsort (Location Area, LA), das aktuelle VLR und MSC. [2]

Zur weltweit eindeutigen Identifizierung eines Teilnehmers und dessen zugehörigen HLR-Eintrags dient die IMSI. Dieser Identifikator des Teilnehmers wird in jeder teilnehmerspezifischen Signalisierung mit übertragen. Durch diese weltweite Eindeutigkeit ist es im GSM-System möglich, dass ein Teilnehmer sein mobiles Endgerät in jedem GSM-Netz, mit welchem sein eigener Provider eine Roaming-Übereinkunft unterhält, benutzen kann. [3]

### *2.2.5 Schnittstellen zwischen den einzelnen Komponenten*

Wie bereits aus Abbildung 2 ersichtlich wird, kommunizieren die verschiedenen Elemente des Kernnetzes über unterschiedliche, standardisierte Schnittstellen (A, E, C, D) miteinander:

- A-Interface: Über das A-Interface sind die unterschiedlichen Feststationssysteme mit dem zuständigen MSC verbunden. Diese Verbindung besteht aus einigen E-1 Verbindungen (in Europa) mit einer Kapazität von jeweils 2 Mbit/s. Eine solche E-1 Verbindung wird über Twisted Pair, Koaxial-Kupfer-Kabel oder Glasfaser aufgebaut.
  Da eine E-1 Verbindung nur 31 Kanäle à 64 kbit/s übertragen kann, ist eine Vielzahl von solchen Verbindungen von Nöten, um die notwendige Bandbreite abzudecken. Über diese Schnittstelle werden sowohl Signalisierungsnachrichten als auch die eigentliche Sprache geleitet.
- E-Interface: Das E-Interface, welches wiederum über gebündelte E-1 Verbindungen verfügt, dient zur Verbindung der verschiedenen MSC miteinander. Über dieses Interface werden die Signalisierungsnachrichten, welche z.B. bei einem Handover zwischen einem MSC und einem anderen auftreten, geleitet. Neben den Signalisierungsaufgaben werden über diese Schnittstelle auch Sprachpfade aufgebaut.
- C-Interface: Mit Hilfe des C-Interfaces werden die MSC mit dem HLR des Mobilfunknetzes verbunden. Über diese Schnittstelle werden im Gegensatz zum A- und E-Interface nur Signalisierungsnachrichten ausgetauscht, welche wieder über E-1 Verbindungen laufen.
- D-Interface: Über diese D-Schnittstelle läuft die standardisierte Kommunikation, basierend auf dem Mobile Application Part (MAP)-Protokoll, zwischen VLR und HLR ab. [3]

## 3. GPRS

Dieser Abschnitt befasst sich mit dem ersten Evolutionsschritt, der im GSM-System vollzogen wurde. Angetrieben wurden die im Folgenden angeführten Neuerungen vor allem dadurch, dass Mitte der 1990er Jahre der Einfluss und die Verbreitung des Internets immer weiter anstieg und man nun auch im Mobilfunk eine effiziente Art und Weise suchte, um dieses auf den mobilen Endgeräten nutzen zu können.

Die bisherige leitungsvermittelte Lösung, die in der Einführung zu GSM erläutert wird, ist optimal für Sprachübertragung, jedoch bringt sie für die Datenübertragung auch einige gravierende Nachteile mit sich. Datenübertragung zeichnet sich nämlich typischerweise dadurch aus, dass man variable bzw. „burst“-Datenraten hat. So werden beim Laden einer Website zunächst in kurzer Zeit viele Daten angefordert, sobald die Übertragung aber abgeschlossen ist und der Benutzer die Information erhalten hat, ist für eine bestimmte Zeit keine Datenübertragung mehr notwendig. Leitungsvermittelte Systeme haben jedoch eine limitierte feste Bandbreite und können somit die Datenraten nicht für andere Teilnehmer freigeben oder individuell und flexibel erhöhen.

Um diese Datenübertragung besser zu unterstützen, wurde mit GPRS die Paketvermittlung als Ergänzung zur Leitungsvermittlung im GSM-Netz eingeführt.

Neben Neuerungen auf der Luftschnittstelle, die vor allem die Übertragungsrate (von 9.6 kbit/s auf bis zu 170 kbit/s) erhöhten und die Verbindungszeit (von bis zu 20 Sekunden auf 5 Sekunden) erniedrigten, brauchte GPRS auch in der Infrastruktur neue Funktionalität. Dazu zählen sowohl neue Funktionalitäten in den BSCs, als auch völlig neue Komponenten in der Kernnetzarchitektur. [3]

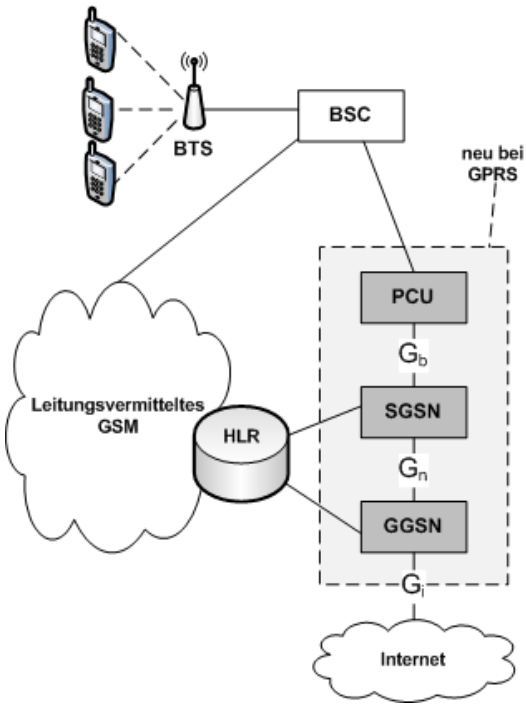



**Abbildung 3 - GRPS Referenzarchitektur**

## 3.1 Packet Control Unit (PCU)

Die leitungsvermittelnden Funktionen im BSC eignen sich nicht um den paketorientierten GPRS Verkehr zu managen. Deshalb wurde mit der PCU eine neue Netzwerkkomponente eingeführt, welche das paketvermittelte Gegenstück zur leitungsvermittelten BSC darstellt. Dies bedeutet, sie regelt den Zugriff der einzelnen mobilen Station auf das Funkmedium. Die Zusammenarbeit von BSC und PCU läuft dann so ab, dass die PCU eine bestimmte Anzahl von Zeitslots von der BSC zur Verfügung gestellt bekommt, die sie dann kontrollieren kann. [3]

## 3.2 Serving GPRS Support Node (SGSN)

Die SGSN kann als das paketvermittelte Pendant zur leitungsvermittelten MSC im ursprünglichen GSM-Netz angesehen werden. Wie in Abbildung 3 ersichtlich ist, ist die SGSN zum mobilen Endgerät hin mit einer (oder mehrerer) BSS über die zugehörige PCU verbunden, in Richtung Internet bzw. zu anderen paketorientierten Netzen mit der Gateway GPRS Support Node (GGSN), welcher im nachfolgenden Abschnitt erklärt wird. Die SGSN transportiert also Pakete, die für einen bestimmten Teilnehmer über die GGSN ankommen, zum richtigen PCU weiter, und empfängt aus der entgegengesetzten Richtung Datenframes von der PCU und leitet sie an die GGSN weiter [3].

Zur Übertragung der Daten im GRPS Kernnetz, also die Schnittstelle $G_n$ zwischen SGSN und GGSN, wird das Internet Protokoll verwendet [2]. Der große Vorteil davon ist, dass IP über alle möglichen Übertragungstechnologien, die in darunterliegenden Schichten vorhanden sind (z.B. ATM, Glasfaser, Ethernet, Mikrowellen), funktioniert. Zur Übertragung der Daten zwischen SGSN und PCU, also über das $G_b$-Interface wird weiterhin das Frame Relay Protocol (FRP) ausgeführt. Man sieht also dass, diesem Protokollstack eine komplexe Kapselung vorhanden.

Eine weitere Neuerung gegenüber der ursprünglichen GSM-Architektur ist, dass der Verschlüsselungsendpunkt nicht wie bei GSM die BTS ist, sondern die Verschlüsselung nun bis zur SGSN reicht.

Neben der Datenweiterleitung und Verschlüsselungsfunktionalität ist die SGSN auch noch für den GPRS Sitzungsaufbau und für die Verwaltung der einzelnen Sitzungen zuständig. Im Grunde bedeutet das nichts anderes, als dass dem Teilnehmer über das Packet Data Protocol (PDP) eine IP-Adresse zugeordnet wird und das Routing beim Roaming des Teilnehmers zu einer anderen BSS angepasst werden muss. [3]

## 3.3 Gateway GPRS Support Node (GGSN)

Wie im letzten Abschnitt beschrieben, ist die SGSN für die Weiterleitung und das Routing der Pakete vom Funknetzwerk zum Kernnetz und umgekehrt zuständig. Die GGSN stellt nun die Verbindungsstelle des GPRS Kernnetzes zu den externen paketbasierten Netzwerken wie dem Internet dar, analog zum G-MSC in der leitungsorientierten Vermittlung.

Weitere Aufgaben sind die eigentliche Zuteilung der IP-Adresse, die dann von SGSN weiter zum mobilen Endgerät geleitet wird. Für die Verbindung des mobilen Teilnehmers hin zum Internet stellt die GGSN eine Art Fixpunkt für die Kommunikation dar. Wenn sich der Teilnehmer bewegt, und sich somit in eine BSS und neue SGSN einbuchen muss, so setzt die GGSN ihre Routen in der Routingtabelle so um, dass ein GPRS Tunnel zur neuen SGSN aufgebaut wird und somit die Pakete dorthin geleitet werden. Dadurch bleibt diese Bewegung sowohl für das Endgerät als auch für die andere Partei der Verbindung im Internet transparent, wodurch die direkte Verbindung ins Internet (in der Theorie) nicht gestört wird [3]. Dieser Tunneling-Mechanismus wird durch das GPRS Tunneling Protocol (GTP) ermöglicht. [3]

# 4. UMTS

UMTS beschreibt das Mobilfunknetz der 3. Generation. Anders als diese Bezeichnung vielleicht vermuten lässt, handelt es sich hierbei nicht um ein komplett neudesigntes System [3], wie dies beim Übergang von den analogen Netzen (1G) zu GSM (2G) der Fall war, sondern baut zu großen Teilen auf der GSM und GPRS Architektur auf. Somit stellt UMTS die nächste Phase der Evolution von GSM dar, welche in mehreren Schritten durchlaufen wird. In der Entwicklung von UMTS spricht man in der Standardisierungskommission 3rd Generation Partnership Project (3GPP) hierbei von Releases, welche in den nachfolgenden Abschnitten mit den entsprechenden Neuerungen chronologisch aufgelistet und beschrieben werden.

## 4.1 UMTS Release 99

Dieses Release stellt den ersten Schritt von UTMS dar. Die hauptsächliche Veränderung im Vergleich zu GSM war ein vollkommen neues Funkzugangsnetzwerk (Radio Access Network, RAN), das sogenannte UMTS Terrestrial Radio Access Network (UTRAN). Es basiert nicht wie GSM auf frequenz- oder zeitbasiertem Multiplexing, sondern auf Code-Multiplexing, dem Wideband Code Division Multiple Access (WCDMA). Dies ermöglicht nun Datenraten von bis zu 384 kbit/s im Downlink und bis zu 128 kbit/s im Uplink, was vor allem darauf abzielt, einen schnellen Internetzugang und somit datenintensive Anwendungen auf dem mobilen Endgerät anbieten zu können.

Dieses UTRAN, besser gesagt eine Basisstation (Node-B genannt) in diesem Funknetz, kann nun mit relativ wenig Aufwand über einen Radio Network Controller (RNC) mit dem bereits bestehenden GSM/GPRS-Kernnetz verbunden werden. Sie stellt aus Sicht des Kernnetzes nur eine weitere Funkschnittstelle dar, welche durch Softwareupdates und Erweiterungskarten im SGSN hinzugefügt werden kann. Im Kernnetz selbst gab es also nur kleinere Veränderungen. Als größte Veränderung wurde das auf dem $G_b$-Interface zwischen SGSN und PCU im GRPS noch vorhandene Frame Relay Protocol durch ATM Technik ersetzt. [3]

## 4.2 UMTS Release 4

Während die Änderungen des Release 99 auf den Funkbereich beschränkt bleiben, beziehen sich die Neuerungen des Release 4 auf den leitungsvermittelten Bereich des Kernnetzes.

Der Status Quo vor diesem Release war der, dass alle leitungsvermittelten Verbindungen über E-1 Verbindungen innerhalb von 64 kbit/s Zeitslots abliefen. Dieser Bereich wird nun durch das Bearer Independent Core Network (BICN) revolutioniert, welches den Verkehr in ATM Zellen oder IP Pakete verpackt, anstatt sie in diesen 64 kbit/s Zeitslots zu kapseln. Um dies zu ermöglichen, muss der MSC in zwei Bereiche aufgeteilt werden: den MSC Call Server und das Media Gateway. Der Call Server übernimmt die Signalisierungsaufgaben für Anrufkontrolle und das Mobilitätsmanagement. Der eigentliche Verkehr aber wird über das Media Gateway geleitet, welches eine Transcodierung der verschiedenen Techniken unternehmen kann. Kommt also beispielsweise ein Sprachanruf über die GSM A-Schnittstelle über einen 64 kbit/s Zeitslot an, so transcodiert das Gateway dies zu ATM oder zu IP-Paketen und leitet den Verkehr in dieser Form weiter zum nächsten Media Gateway. Somit erreicht man im eigentlich Kernnetz eine

Architektur, in der die Datenübertragung nur noch auf ATM oder IP Basis abläuft. Dieses Release wurde vor allem von den Mobilfunkanbietern gefordert und unterstützt, weil sie auf diesem Weg Kostenersparnisse im Kernnetz-Backbone erreichen können. [3] Diese Ersparnis ist dadurch zu erklären, dass der Provider seine paketorientierte Vermittlungsarchitektur neben der Datenübertragung auch für Sprachanrufe nutzen kann. Somit braucht es keine zusätzliche Erweiterung der „alten" Leitungsvermittlung mehr, um auf steigende Gesprächszahlen zu reagieren.

Der Aufbau eines solchen Release 4 basierten Netzwerks ist in Abbildung 4 im oberen Teil ersichtlich. PSTN symbolisiert dabei das öffentliche Festnetz, HSS steht für das HLR (s. Release 5) [3].

## 4.3 UMTS Release 5

Das Release 5 treibt die Änderungen im Kernnetz weiter voran, und definiert eine rein auf IP basierende Architektur (end-to-end all-IP Network), d.h. alle bisherigen leitungsvermittelten Komponenten wie z.B. der MSC verschwinden aus einem reinen Release 5 UMTS-Netz. Der untere Teil der nachfolgenden Abbildung 4 zeigt den Aufbau eines solchen Kernnetzes.

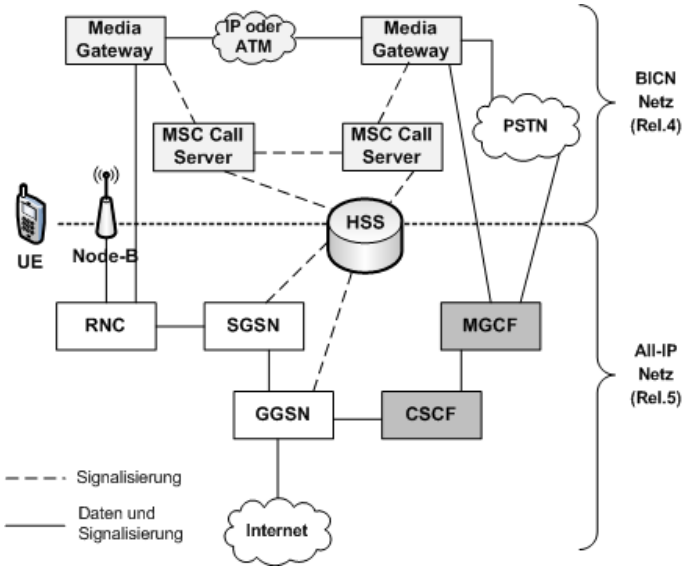



**Abbildung 4 - Release 4 Netz (oberer Teil) und Release 5 Netz (unterer Teil)**

Der MSC wird in dieser Architektur durch das IP Multimedia Subsystem (IMS) ersetzt. Den Kern dieses IMS stellt eine Reihe von Komponenten dar, welche die Call Session Control Function (CSCF) ausüben. CSCF ist im Wesentlichen eine SIP (Session Initiation Protocol) Architektur, wie sie auch in der Festnetztelefonie für VOIP (Voice over IP) eingesetzt wird. Durch diese neue Architektur werden also Sprachanrufe, die rein auf IP-Basis beruhen, möglich. Sind beide Parteien bereits in diesem von Release 5 definierten All-IP-Netz, so werden die gesamten Daten des Anrufs von Ende-zu-Ende direkt mit IP transportiert. Das CSCF fungiert nur noch als Anrufaufbau- und Anrufkontrollorgan. Ist ein Teilnehmer noch über ein leitungsvermittelndes Netzwerk angebunden, so wandelt die Komponente der Media Gateway Control Function (MGCF) die IP Pakete in entsprechende leitungsvermittelnde Dateneinheiten um.

Diese Art der vollkommen auf IP basierten Kommunikation wird deshalb möglich, da in Release 5 und Release 6 auch auf der Funkschnittstelle weitere Verbesserungen eingeführt werden, welche unter dem Begriff High Speed Packet Access (HSPA) zusammengefasst werden. Im Detail werden diese Verbesserungen durch den High Speed Downlink Packet Access (HSDPA) bzw. High Speed Uplink Packet Access (HSUPA) beschrieben, und ermöglichen eine deutlich höhere Bandbreite auf der Luftschnittstelle. Diese ist für die VOIP-Kommunikation auch notwendig, da ein Sprachanruf über IP deutlich bandbreitenintensiver ist als sein leitungsorientiertes Pendant. Grund hierfür ist vor allem die Tatsache, dass durch die Verwendung von IP in der Sprachübertragung ein sehr großer Header-Overhead entsteht, der ca. 50% der gesamten benötigten Bandbreite ausmacht. [3]

# 5. LTE und SAE (EPS)

Das nachfolgende Kapitel beschreibt den nächsten großen Entwicklungsschritt der Mobilfunkkommunikation, das Evolved Packet System (EPS), welches der Nachfolger von UMTS werden soll.

## 5.1 Einführung und Designziele

EPS selbst beschreibt dabei keine wirkliche Architektur bzw. kein System, sondern ist nur ein Sammelbegriff für das „Long Term Evolution" (LTE)-Programm und die Service Architecture Evolution (SAE). LTE beschreibt dabei eine Initiative, welche das Neudesign des Funkzugangsnetzes des Mobilfunknetzes verfolgt. Das SAE-Programm hingegen befasst sich mit der Weiter- bzw. Neuentwicklung des Kernnetzes. [4]

Durch diese zwei neuen Ansätze und Architekturen sollen folgende Designziele erreicht werden:

- *Paketvermittlung*: Jegliche Kommunikation im Mobilfunksystem wird paketorientiert durchgeführt (wie es auch schon ab UMTS Release 5 vorgesehen ist) [3].
- *Schnellere Statuswechsel:* In den bisherigen paketorientierten Netzen (z.B. HSPA) dauert es im Gegensatz zu drahtgebundenen Netzes sehr lange, bis das mobile Endgerät nach einer bestimmten Zeit der Inaktivität wieder mit dem Netz verbunden ist und die erste Nutzdaten übertragen werden können. Diesen sich in Hinblick auf die Nutzbarkeit des Systems negativ auswirkenden Punkt, will man mit EPS verbessern, indem man die maximale Zeit, die ein Statuswechsel vom „Leerlauf" der Verbindung bis hin zur vollständigen Funktionsfähigkeit dauert, auf unter 100 ms begrenzt [4].
- *Latenzminimierung*: Eine weitere sich negativ auf die Verwendbarkeit auswirkende Eigenschaft von UMTS und GPRS-Netzwerken ist die hohe Übertragungsverzögerung. So muss man in heutigen HSPA-Netzwerken mit einer Latenz von über 50 ms rechnen (basierend auf Messungen), was vor allem in Anwendungen mit Echtzeitanforderung (Telefonie, Echtzeit-Spiele) nicht akzeptabel ist. In LTE wurde nun entschieden, dass diese Übertragungsverzögerung höchstens 5 ms betragen darf, um die Wartezeiten denen in drahtgebundenen Netzwerken anzugleichen.
- *Flexible Bandbreite:* Die Bandbreite von HSPA-Netze ist zurzeit auf 5 MHz fixiert. Es kann für manche Anwendungsszenarien jedoch sinnvoll sein, die Bandbreite zu reduzieren oder um höheren Durchsatz zu erreichen diese auch zu erhöhen. Somit hat man in LTE die „skalierbare Bandbreite" eingeführt, welche genau diese Anpassung an die verschiedenen Szenarien ermöglichen soll.
- *Hoher Durchsatz:* Auf der neuen LTE-Funkschnittstelle soll unter idealen Bedingungen ein Durchsatz von 100 Mbit/s erreicht werden. [4]

## 5.2 Roadmap

Die nachfolgende Abbildung 5 zeigt den „Fahrplan“ für das EPS. Hier wird ersichtlich, dass die Entwicklung der neuen LTE-Funkschnittstelle der erste Schritt hin zu einem neuen Mobilfunkstandard war. Dies ist auch der Grund, dass bis heute LTE (und nicht EPS) der gebräuchliche Name für die gesamte neue Architektur ist. In dieser Arbeit wird LTE jedoch zur expliziten Unterscheidung nur für die Neuerungen auf der Funkschnittstelle verwendet.

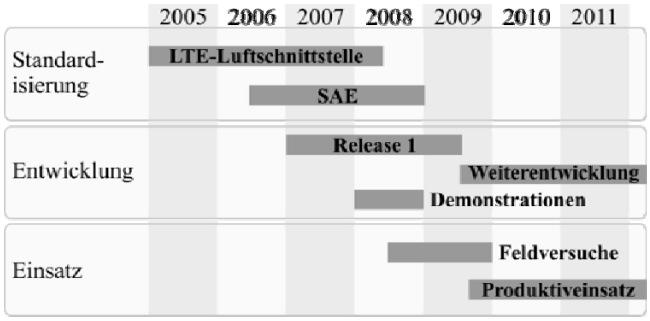



**Abbildung 5 - EPS roadmap [5]**

## 5.3 Architektur

In diesem Kapitel werden nun schrittweise die neuen Komponenten und Mechanismen der EPS-Architektur (siehe Abbildung 6) beschrieben, durch welche die Designziele aus Abschnitt 5.1 erreicht werden können.

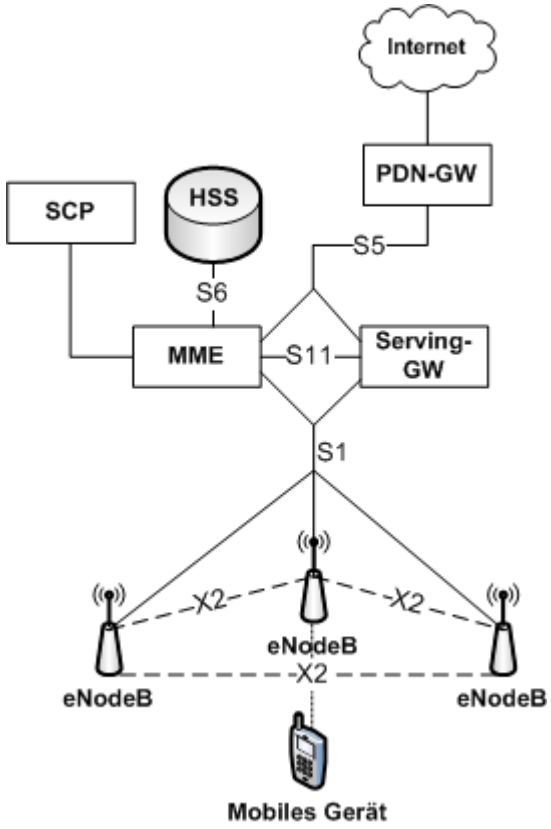



**Abbildung 6 - LTE-Netzwerk**

### *5.3.1 Verbesserte und erweiterte Basisstationen*

Verglichen mit dem Funknetzwerk von UMTS ist das neue LTE-Funknetzwerk „schlanker“ und weniger komplex, dabei jedoch mit mehr Funktionalität und Kompetenz ausgestattet geworden. So wurde im Design festgelegt, dass der RNC, der im UMTS-Netz noch als Verbindungsstelle zum Kernnetz diente und für viele Verwaltungsaufgaben zuständig war, im LTE-Netz nicht mehr vorhanden ist und seine Funktionalität teilweise ins Kernnetz und teilweise direkt in die LTE-Basisstationen gezogen wird. Zur Unterscheidung der LTE-Basisstationen von denen von UMTS werden diese eNodeB (Enhanced NodeB) genannt.

Die eNodeBs führen also nun das gesamte Verkehrs-, Power Loop- und Anrufmanagement für die Luftschnittstelle eigenständig durch. Da auch der Handover zwischen zwei eNodeBs nun von den beteiligten Basisstationen selbst ausgehandelt werden muss, können diese über die X2-Peer-to-Peer-Schnittstelle direkt miteinander verbunden werden. Über diese X2-Verbindung können dann außer den Kontrollnachrichten auch Benutzernutzdaten laufen. Dadurch wird der Paketverlust bei einem Handover auf ein Minimum reduziert. Dieses X2-Interface ist allerdings optional, denn die eNodeBs können auch noch wie in klassischen UMTS-oder GSM-Netzen über das Kernnetz kommunizieren, um den Handover einzuleiten. Durch diese Verbindung über das Serving-Gateway (Serving-GW) kann es aber dazu kommen, dass Pakete während des Handovers verloren gehen. Ein solcher Paketverlust tritt genau dann auf, wenn Pakete vom Kernnetz noch zur ursprünglichen eNodeB gesendet werden, diese in der Zwischenzeit aber bereits den Handover zur nächsten Station abgeschlossen oder den Funkkontakt zum mobilen Gerät verloren hat. Verschärft wird diese Situation noch dadurch, dass in LTE-Netzen im Gegensatz zu GSM/UMTS nur ein Hard-Handover möglich ist, d.h. eine mobile Station kann immer nur mit einer eNodeB verbunden sein, und nicht wie bei GSM/UMTS mit mehreren.

Eine weitere Neuerung ist die Art der Schnittstelle (S1) welche die eNodeB mit dem Kernnetz verbindet. Im Gegensatz zu UMTS-Netzen, bei denen diese Verbindung noch über langsame ATM-Netze (2 Mbit/s) lief, beruht sie bei LTE vollkommen auf Ethernet-Technologie, über welche dann der Datentransport durch IP stattfindet. Dadurch sind die eNodeBs über schnelle Gigabit-Verbindungen angebunden, was aufgrund der theoretischen 100 Mbit/s über die Funkschnittstelle auch notwendig ist. [4]

### *5.3.2 Kernnetz*

Die Abbildung 6 zeigt, dass die eNodeBs nicht wie bisher in GPRS/UMTS-Netzen nur mit einer Komponente, nämlich der SGSN, verbunden sind. Es existieren nun zwei logische Entitäten: die Mobility Management Entity (MME) und das Serving-Gateway (Serving-GW). Diese logische Trennung der SGSN in zwei Komponenten wurde aufgrund der besseren Skalierbarkeit eingeführt. Weitere Elemente des Kernnetzes sind das Packet Data Network (PDN) und der Home Subscriber Server (HSS). [4]

Die Aufgaben sind dabei wie folgt verteilt:

#### *5.3.2.1 MME*

Die MME dient als Kontrollzentrum. Sie ist somit für die Teilnehmerauthentifizierung, das Sitzungsmanagement sowie die Handover-Unterstützung zwischen eNodeBs und von/zu anderen Funknetzwerken, z.B. GSM oder UMTS, zuständig. Weitere Aufgaben sind das „Location tracking“ eines mobilen Geräts auch dann, wenn keine aktive Verbindung besteht und die Auswahl des Internet-Gateways, wenn der mobile Teilnehmer eine Verbindungsaufbau einleitet und somit eine IP-Adresse beziehen will. [4]

#### *5.3.2.2 Serving-GW*

Das Serving-GW führt den eigentlichen Transport der Nutzdaten durch, d.h. über dieses Gateway laufen alle Pakete vom mobilen Endgerät ins Internet und umgekehrt. Wie bereits bei GPRS eingeführt und bei UMTS übernommen, werden auch im LTE-Netz IP-Tunnel, basierend auf dem GTP, zwischen dem Funknetzwerk und dem Kernnetz aufgebaut. Dadurch erreicht man die bereits beschriebenen flexiblen Wege. Da in LTE-Netzen keine RNCs mehr existieren, endet der Tunnel direkt in der eNodeB.

Wie Abbildung 6 zeigt, sind sowohl MME als auch Serving-GW über dieselbe S1-Schnittstelle mit den eNodeBs verbunden. Um die Kontrollnachrichten von den Nutzerdaten zu trennen, gibt es als Unterscheidungsmöglichkeit das S1-C-Protokoll für die Signalisierungsnachrichten und das S1-U-Protokoll für den Aufbau der Tunnel für die Nutzdaten.

Anders als bei GRPS/UMTS, ist bei EPS nicht für jede Basisstation (bzw. BSC) genau eine SGSN zuständig, sondern es ist eine sogenannte „Mesh"-Struktur möglich. Dies bedeutet, dass mehrere, beliebig viele MMEs und Serving-GWs mit einer eNodeB verbunden sein können. Somit wird zum Einen die Ausfallwahrscheinlichkeit des Netzes reduziert, weil beim Ausfall einer MME eine weitere einspringen kann (wenn dies vom Provider entsprechend konfiguriert wurde). Zum Anderen sinkt somit auch die Zahl der Handovers die zwischen den MMEs durchgeführt werden müssen. [4]

#### *5.3.2.3 PDN-Gateway*

Das PDN-GW im EPS ist das Gegenstück zum GGSN in GRPS/UMTS-Netzen. Seine Hauptaufgaben sind somit die Zuteilung von IP-Adresse an die mobilen Teilnehmer und das Verbergen der Mobilität dieser mobilen Endgeräte durch die bereits beschriebenen Tunneling-Mechanismen. Die Verbindung zu MME und Serving-GW erfolgt dabei über die S5-Schnittstelle, auf der das GTP-U (GTP-user) Protokoll als auch das GTP-S (GTP-signaling) ausgeführt wird. GTP-U tunnelt dabei den Benutzerverkehr, GTP-S dient zum Aufbau und nachfolgender Veränderungen dieses Tunnels. [4]

#### *5.3.2.4 HSS*

Eine weitere wichtige Komponente der EPS-Architektur ist das HSS, welcher das Home Location Register (HLR) aus den vorherigen Mobilfunkgenerationen darstellt. Das HSS ist also ein erweitertes HLR und enthält Teilnehmerinformationen für GSM, GPRS, UMTS, LTE und IMS.

Das HSS ist über das S6-Interface mit den MMEs verbunden. Anders als bei UMTS wird bei der Übertragung der Daten über diese Schnittstelle nicht das MAP-Protokoll eingesetzt, sondern das IP-basierte Diameter-Protokoll verwendet. Da das HSS jedoch auch abwärtskompatibel zu GSM und UMTS ist, muss natürlich auch das MAP weiterhin implementiert sein. [4]

#### *5.3.2.5 Zusammenspiel mit GSM/GPRS/UMTS*

Wie bereits im Abschnitt zum HSS angedeutet, wird ein LTE-Netz meist nicht als ein alleinstehendes System implementiert, sondern muss in die bestehende Mobilfunkinfrastruktur eines Providers integriert werden. Dies bedeutet, die Mobilfunkteilnehmer müssen zwischen LTE-, UMTS- und GSM-basierenden Netzen flexibel wechseln können. Dies ist vor allem in der Anfangsphase von LTE sehr wichtig, da hier die Abdeckung durch LTE-Netze erfahrungsgemäß noch sehr beschränkt ist.

Dieser Handover zwischen den verschiedenen Mobilfunkgenerationen erfolgt dabei folgendermaßen:

1. Wenn sich der mobile Teilnehmer an den Rand der LTE-Abdeckung bewegt und eine passende UMTS (oder GSM)-Zelle findet, in die er sich einbuchen möchte, sendet er diesen Handover-Wunsch zu seiner LTE eNodeB.
2. Die eNodeB leitet diesen Bericht dann an die zuständige MME weiter.
3. Die MME kontaktiert die für diese UMTS (oder GSM)-Zelle zuständige 3G (oder 2G) SGSN und fordert die Ausführung des Handovers an.
   Diese Prozedur läuft über das S3-Interface ab. Das ausgeführte Protokoll basiert dabei auf dem bestehenden inter-SGSN Protokoll, welches auch in normalen UMTS (oder GPRS)-Netzen verwendet wird, um die Verantwortung für einen mobilen Teilnehmer von einer SGSN an eine andere zu weiterzugeben. Somit braucht es bei den 3G (bzw. 2G)-Komponenten keinerlei Änderungen.
4. Sobald das 3G (bzw. 2G)-Netzwerk für den Handover bereit ist, sendet die MME einen Handover-Befehl über die eNodeB zum mobilen Gerät.
5. Sobald der Handover vollzogen ist, werden die Nutzdaten für den mobilen Teilnehmer vom LTE Serving-GW nicht direkt über die eNodeB an ihn geschickt, sondern über die zuständige 3G (oder 2G) SGSN geroutet. Das bedeutet also, dass das Serving-GW bei der Übertragung der Daten involviert bleibt. Aus Sicht der SGSN stellt das Serving-GW somit ein GGSN dar.
   Die MME hingegen zieht sich aus dem Teilnehmermanagement zurück, da dieses nun wieder von der SGSN übernommen wird. [4]

### *5.3.3 Weitere Neuerungen durch EPS*

In den vorherigen Abschnitten wurden die Neuerungen der EPS-Architektur aufgezeigt und die verschiedenen Komponenten und ihr Zusammenspiel auf der Funkschnittstelle und im Kernnetz vorgestellt.

Neben den Neuerungen von EPS, die in den Designkriterien aufgezeigt werden, besteht auch noch ein weiterer großer Unterschied zwischen UMTS und EPS. Da UMTS und die zugehörigen mobilen Endgeräte immer noch hauptsächlich für die Übertragung von Sprachanrufen konzipiert sind, ist es dort sinnvoll, erst bei einer wirklich anstehenden Datenübertragung eine IP-Adresse anzufordern, denn Sprachanrufe sind ja auch ohne zugeteilte IP-Adresse möglich.

In LTE-Netzen hingegen ist dies komplett anders. Verbinden sich ein LTE-fähiges mobiles Gerät mit dem LTE-Netz, so muss es eine IP-Adresse anfordern, da jegliche Kommunikation über IP abläuft. Das heißt, ein LTE-Gerät ohne IP-Adresse ist nicht kommunikationsfähig.

Dieses bereits aus anderen Netzen wie LAN/WLAN bekannte Prinzip ist im Bereich des Mobilfunks eine große Neuerung. [4]

## 6. Zusammenfassung und Ausblick

Die in dieser Arbeit beschriebene Evolution der Kernnetze im Mobilfunk macht klar, in welche Richtung sich der Mobilfunk aus Sicht der 3GPP und der Provider bewegen muss. Es geht weg von sprachzentrierter Telefonie, hin zum IP-basierten, paketorientierten Internetverkehr. Während im Bereich der Festnetztelefonie bereits vor Jahren dieser Weg hin zu rein IP-basierten Netzen eingeschlagen wurde, zeichnet sich dieser Trend nun auch im Mobilfunkbereich ab. Mit den UMTS Releases 5 und 6 wurde schon der erste Schritt in diese Richtung getan, mit den kommenden EPS-Netzen wird dieser dann vorgeführt und

vollendet. Wenn die EPS-Netze das halten, was in den Designkriterien versprochen wird, so ist man auf einem guten Weg die Qualität und Bedienfreundlichkeit des mobilen Internets mit dem der drahtgebunden Netze auf eine Stufe zu bringen, sodass die EPS-Netze dieselbe große Verbreitung erlangen wie es GSM im Sprachbereich momentan bereits der Fall ist.

# 7. Literaturverzeichnis


[1]. **GSM Association.** Mobile World Celebrates Four Billion Connections. [Online] 11. 02 2009. [Zitat vom: 16. 05 2009.] http://www.gsmworld.com/newsroom/press-releases/2009/2521.htm.

[2]. **Schiller, Jochen.** *Mobilkommunikation.* München : Pearson Studium, 2003. Bde. 2., überarbeitete Auflage. 3-8273-7060-4.

[3]. **Sauter, Martin.** *Communication Systems for the Mobile Information Society.* West Sussex : Wiley, 2006. 0-470-02676-6.

[4]. —. *Beyond 3G - Bringing networks, terminals and the web togheter.* West Sussex : Wiley, 2009. 978-0-470-75188-6.

[5]. **Motorola, Inc.** Driving 4G: WiMAX & LTE. [Online] [Zitat vom: 05. 06 2009.] http://www.motorola.com/staticfiles/Business/Solutions/Industry%20Solutions/Service%20Providers/Wireless%20Operators/Wireless%20Broadband/wi4%20WiMAX/_Document/StaticFile/a%20Driving_4G__WiMAX_and_LTE.pdf.

[6]. **Sesia, Stefania, Toufik, Issam und Baker, Matthew.** *LTE - The UMTS Long Term Evolution - A Pocket Dictionary Of Acronyms.* s.l. : Wiley.

# Wie verändert man das Internet ?

Tobias Ladurner
Betreuer: Marc Fouquet
Seminar Innovative Internettechnologien und Mobilkommunikation SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: ladurnet@in.tum.de

## Kurzfassung
Das Internet hat bereits viele Veränderungen durchgemacht, vor allem in seiner Anfangsphase als ARPAnet. Jedoch die meisten Änderungen an Technologien sind aus Notwendigkeit heraus entstanden und bilden bis heute unter anderem die Kernprotokolle des Internets. Nach dem großen Durchbruch des Internets mit dem Web wurden zahlreiche Technologien entwickelt um das Internet zu verbessern, aber es war bereits nicht mehr einfach das etablierte Internet zu verändern. Viele Entwicklungen konnten bis heute das Internet nicht erobern. Dieses Paper wird sich mit den technologischen Veränderungen des Internets und den Technologien, die das Internet verändern wollten und es noch vorhaben auseinandersetzen.

## Schlüsselworte
NCP, TCP, IPv6, IP Multicast, Mobile IP, ECN, host.txt, DNS, CIDR, NAT

## 1. EINLEITUNG
Die Umstellung von IPv4 auf Ipv6 müsste jedem bekannt sein, der sich näher mit den Internettechnologien beschäftigt. Seit vielen Jahren tauchen vereinzelt Hinweise auf einen langsamen Wandel auf. In Betriebssystemen sind IPv6 Einstellungen zu finden und der eine oder andere Router hat IPv6 natürlich eingebaut. Dass der Wandel erstaunlich langsam, trotz dessen Notwendigkeit vorangeht, wird im heutigen etablierten Internet als verständlich betrachtet. Jedoch nicht immer war es so schwer das Internet zu verändern. In der Anfangszeit des Internets, als es noch als ARPAnet bekannt war und nur wenige amerikanische Universitäten umfasste, gab es viele Veränderungen, die in kurzer Zeit umgesetzt werden konnten. Vielleicht war die eine oder andere Lösung zu weit vorausschauend (z.B. zu viele Adressen für IPv4), sonst wäre schon viel früher die Notwendigkeit einer Verbesserung aufgetaucht, andere Lösungen wiederum vielleicht einen Tick zu provisorisch. Mit den Nachteilen der Lösungen müssen wir heute noch leben.

## 2. FRÜHE VERÄNDERUNGEN
Das ARPAnet war das erste Paket-vermittelnde Netzwerk. In den 60ern wurde das Projekt gegründet, um ein ausfallsicheres dezentrales Netzwerk zu errichten. Unter anderem für militärische Zwecke im Kalten Krieg, wofür sich später aber das MILnet vom ARPAnet abspaltete. Das ARPAnet erstreckte sich in seiner Anfangszeit in erster Linie zwischen verschiedenen amerikanischen Universitäten. Die Verbindungen wurden mit IMP ( Interface Message Processor ) [3] aufgebaut, das Äquivalent zu heutigen Routern. Als erste Anwendungen des ARPAnet entstanden Telnet, FTP, Mail und erste Versuche mit Voice over IP .

In der späteren Phase des ARPAnet, als viele Probleme teilweise mit Notlösungen versehen wurden, dachte niemand, dass diese Protokolle Grundlagen des weltweiten Internets werden. Vielmehr spekulierte man darauf dass OSI Protokolle entwickelt würden um eine effiziente Kommunikation zu gewährleisten. OSI Protokolle trennen sauber die verschiedene Schichten der Kommunikation wie man in Figur 1 sehen kann.

### 2.1 Von NCP zu TCP/IP
Alle Verbindungen wurden zu jener Anfangszeit über NCP (Network Control Program) [4] aufgebaut. NCP baute alle Verbindungen auf, kontrollierte den Paketfluss und verwaltete die Adressierung. Es gab noch keine Trennung der Transport und Netzwerkschicht. Das Protokoll war komplex aufgebaut und auf die Hardware jener Zeit, die IMPs angepasst. Ein eigener Kontrollkanal wurde für jede Verbindung aufgebaut.

Schnell wurde ersichtlich, dass ein flexibleres Protokoll nötig ist um mit einem größeren Netzwerk ( ein paar tausend Hosts ) zu skalieren. Mehrere Protokolle standen zur Auswahl: XNS, Delta-t und TCP ( Transmission Control Protocol ) [5]. Anfangs war in TCP Transport und Netzwerkschicht noch nicht getrennt. Das DoD ( Department of Defense ) entschied sich für TCP. Jedoch ziemlich schnell war klar, dass Adressierung in einem eigenen Protokoll effizienter ist. 1982 begann man TCP zu implementieren. Innerhalb eines Jahres wurden die gesamte Hardware umgestellt. 1983 war ein sogenannter „Flag Day“ an dem 400 Knoten, der Großteil des ARPAnet umgestellt wurden. Die IMP wurden durch neue Standard Interfaces ersetzt.

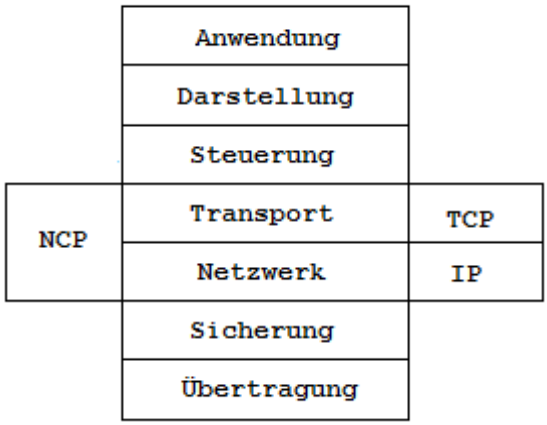



Bild 1: OSI Modell
( 1983 standartisiert )

## 2.2 Von host.txt zu DNS

Seit Anfang des ARPAnet wurde in einer Textdatei (host.txt )vom Network Information Center ( NIC ) für jeden Host die zugehörige Adresse gespeichert. Dies vereinfachte die Verwaltung der wenig benutzerfreundlichen Zahlenadressen. Alle paar Wochen wurde die aktuelle Version der Textdatei heruntergeladen. Mit der Zeit wurde aus ein paar Wochen, jede Woche bis schließlich jeden Tag die aktuelle Textdatei heruntergeladen wurde. Die Datei wuchs mit der Zeit und wurde schwer zu handhaben. Schnell wurde klar, dass es einen Dienst benötigt der die Namenszuweisung zu Adressen automatisiert.
DNS ( Domain Name System ) [6] ermöglicht eine automatische Namensauflösung durch einen auf vielen Servern verteilten hierarchischen Verzeichnisdienst. Root-Nameserver verwalten und speichern die Top Level Domains (z.B. de, org, net, com ) und dienen als zentrale Verwaltungsstellen des verteilten DNS Systems. Durch seine hohe Zuverlässigkeit und Flexibilität wurden bald alle alten Datenbestände in DNS integriert. Heutzutage werden alle Internetadresse durch DNS verwaltet. Erst durch DNS konnte das Web entstehen mit seinen unzähligen Links wie wir es heute kennen.

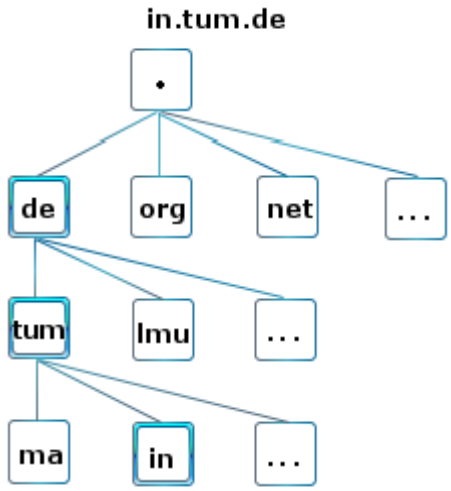



Bild 2: DNS Hierarchie

## 2.3 TCP Staukontrolle

In der Mitte der 80er, als das Netzwerk unter Vollbetrieb lief, aber immer noch ohne Verrichtung von nützlicher Arbeit die über die Forschung hinausging, gab es komplette Zusammenbrüche des Internets. Das war das erste mal dass das Internet zusammenbrach, bevor eine Änderung durchgeführt wurde. Der Grund der Zusammenbrüche lag in der Retransmission Strategie von TCP. TCP beherrschte zwar Flusskontrolle und verhinderte, dass der Empfänger überlastet wurde, aber nicht, dass es innerhalb des Netzwerkes zu einem Datenstau kommt. Das Internet wurde durch die mittlerweile größere Anzahl der Nutzer teilweise unbenutzbar. Ein Lösung wurde dringend gesucht.

Verbindungsorientierte Kommunikation hat sich zu jener Zeit schon etabliert und man wollte nicht darauf verzichten. Ein Vorschlag von Van Jacobson bestand darin eine Staukontrolle[7] in TCP einzubauen. Die Staukontrolle bestand darin den Datenfluss langsam zu vergrößern bis ein Stau entsteht, dann wird der Datenfluss wieder verkleinert und das Spiel beginnt von neuem. Der Vorschlag war eine schnelle Lösung und wurde bald umgesetzt um das Netz wieder funktionstüchtig zu bekommen.

Es wird spekuliert ob es irgendwann wieder zu einem Zusammenbruch des Internets kommen kann, Datenstaus sind nicht vollständig verhindert, zumal es nicht nur TCP als Transportprotokoll gibt. Eine flexiblere Lösung wäre wohl auf jeden Fall eine eigene Staukontroll-Schicht gewesen. Das heute vergleichsweise schnelle Internet hat sich aber bis heute als relativ stabil behauptet.

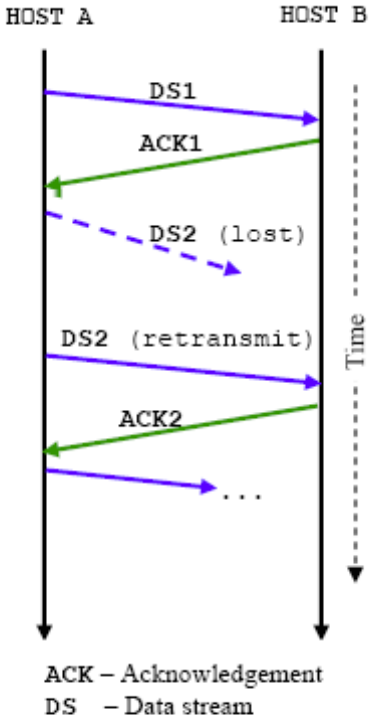



Bild 3: TCP Retransmission

## 2.4 CIDR Klassenlose Adressierung

Adressen können in Netzwerkadresse und Hostadresse gtrennt werden. Die ursprüngliche Internetadresse beherrschte 3 Klassen um Netzwerke zu adressieren: A mit 16M, B mit 65k und C mit 256 Hostadressen. Anhand der IP Adresse konnte man automatisch sehen, in welche der 3 Klassen die Adresse eingeordnet werden kann. Während A für die meisten Organisationen zu große war und C zu klein wurden hauptsächlich Netzwerke der Klasse B benutzt. Meistens wurden aber nicht alle 65K Adressen im Netz vergeben, ein Großteil der Adressen lag brach. Die Klasse B Netzwerke wurden langsam knapper und die steigende Anzahl von Klasse C Netzwerke als Ersatz drohten die globalen Routingtabellen zu überfordern.

| | Präfix | Adresse* | Netzmaske | Hosts |
|---|---|---|---|---|
| **Klasse A** | 0... | 0.0.0.0 | 255.0.0.0 | 16.777.214 |
| **Klasse B** | 10... | 128.0.0.0 | 255.255.0.0 | 65.534 |
| **Klasse C** | 110.. | 192.0.0.0 | 255.255.255.0 | 254 |

* Start des Adressbereichs
Bereiche laufen bis kurz vor der nächsten Klasse. Klasse C läuft bis 239.255.255.255. Zwischen 224.0.0.0 und 255.255.255.255 liegt noch Klasse D und E die damals für Multicast und Ausnahmen reserviert wurden.

1985 wurde Subnetting[9] als erste Lösung eingeführt. Subnetting ermöglicht es Subnetze einzubauen und somit einen Hierarchischen Aufbau eines Netzwerkes zu realisieren. Die Subnetzwerke sind von den globalen Routern nicht sichtbar somit werden sie entlastet und Organisationsinterne Router übernehmen das weitere zerlegen der IP. Mit Hilfe von Netzmasken konnte man festlegen, welche Bits Teil der Netzwerkadresse sind und welches Bit zur Hostadresse gehörte ( z.B. 255.255.0.0 für Klasse B Adressen ). Die Adressen wurde jedoch immer noch global gesehen in Klassen eingeteilt.

Schließlich wurde 1992 Supernetting[10] und 1993 CIDR ( Classless Inter-Domain Routing ) eingeführt, dass eine klassenlose Adressierung ermöglicht. Jede Netzwerkadresse kann genau auf die Bedürfnisse einer Organisation angepasst werden. In CIDR Schreibweise wird die Länge der Netzwerkadresse nach der IP Adresse mit einem Schrägstrich angegeben. Subnetting, Supernetting und CIDR wurden außerdem bereits entwickelt um einen drohenden Mangel an frei verfügbaren IPv4 Adressen hinauszuschieben.

# 3. NEUE TECHNOLOGIEN

Mit der Entwicklung des ersten Webbrowsers NCSA Mosaic begann das Zeitalter des Web und des Internets wie wir es heute kennen. Firmen entstanden rund um das Internet. Aus einem wissenschaftlichen Netzwerk wurde ein kommerzielles Netzwerk für die breite Masse. Wo man Anfangs Probleme hatte sinnvolle Anwendungen für das ARPAnet zu finden so entstanden für das Internet unzählige Anwendungen, die bis heute die Welt verändert haben. War das Netz am Anfang statisch und kommerzielle Webseiten Seite dominierten den Blick auf das Web so veränderte sich bis heute das Netz zu einer interaktiven Plattform einer globalisierten Netzgesellschaft. Blogs, Foren, Chats, Videoportale, Online Spiele, soziale Netzwerke, Wikipedias, Internet-Versandhandel, VoIP und vieles mehr bilden heutzutage das Bild des Internets.

Entwicklungen gab es seit Anfang des Webs viele, Probleme des Internets waren bekannt wie zum Beispiel der schrumpfende Anteil an freien IPv4 Adressen. Jedoch es war nicht mehr leicht Änderungen durchzusetzen. Die Kernprotokolle waren etabliert und auf millionenfacher Hardware implementiert. Die meisten Technologien jener Zeit konnten sich nur in kleinen Netzwerken durchsetzen. Andere wiederum sind aus Sicherheitsgründen bei vielen Servern und Providern gesperrt. Manchen fehlt auch einfach deren Unterstützung und nur wenige inkompatible Server bringen das Protokoll zum erliegen.

## 3.1 IP Multicast

IP Multicast hat viel versprechend in den frühen 90ern eine Möglichkeit aufgezeigt um effizienter Medien ( wie z.B. TV und Radio ) im Internet zu verbreiten. Mit IP Multicast wäre es möglich Datenpakete an ein ausgewähltes Publikum zu senden ohne für jeden Benutzer redundante Pakete zu versenden. Bei den Routern werden die Pakete für die einzelnen Benutzer vervielfältigt.

Leider gib es einige Probleme hinsichtlich der Kostenmodelle und der Sicherheit im Internet. Mit IP Multicast würde die Last bei Routern steigen, da sie die Pakete nicht nur weiter liefern, sondern ebenfalls vervielfältigen müssten. Das größte Problem steht aber hinsichtlich der bestehenden Kostenmodelle. Provider verkaufen Internetzugänge in erster Linie mit der Bandbreite, die ein Anschluss gewährt, denn die Last auf Server im Internet steigt proportional zum Datenverkehr den ein Host verursacht. Jedoch mit IP Multicast wäre es möglich aus einem geringen Anfangsdatenstrom einen gewaltigen Datenfluss im Internet zu erzeugen und eine große Last auf den Servern zu erzeugen. Aus dem gleichem Argument entsteht auch ein Sicherheitsrisiko. Jeder DoS ( Denial of Service ) Angriff wird durch die Bandbreite der benutzten Hosts begrenzt, aber mit IP Multicast wäre diese Begrenzung entfernt und Serverfarmen könnten massive Angriffe hervorrufen die das Internet lahmlegen könnte. Aus diesen Gründe konnte sich IP Multicast noch nicht im Internet durchsetzen auch wenn es im Moment hinsichtlich IPTV ein Interesse gibt und gebe wird Lösungen für IP Multicast zu finden und es zu implementieren.

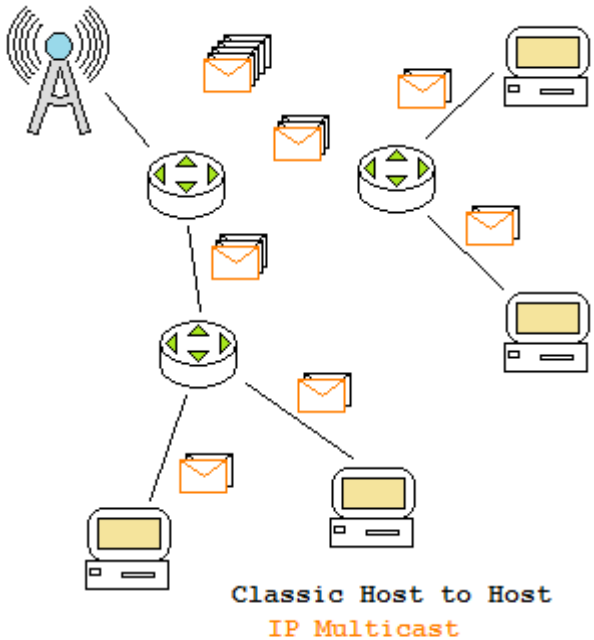



Bild 4: IP Multicast

## 3.2 ECN

ECN ( Explicit Congestion Notification ) ist ein Verfahren um die bestehenden Staukontrolle zu verbessern. Anhand von zwei Bit in TCP und IP Header können Router eine mögliche Überlast erkennen und den Datenfluss reduzieren. Anstatt ein Paket bei einem Anzeichen einer Staubildung zu verwerfen wird es markiert und damit dem Sender und Empfänger mitgeteilt die Datenrate zu verringern. Dadurch kann bei geringem Stau ein Paketverlust vermieden werden und die Paketverzögerung wird durch kurze Warteschleifen verringert. Außerdem kann ein TCP Endpunkt erkennen ob die Pakete aufgrund Verbindungsfehler oder durch die Staukontrolle verloren gingen.

In den 90ern wurde ECN standardisiert, aber bis heute ist ECN noch nicht ausreichend verbreitet. Linux Kernel ab 2.4 und viele neuere Router unterstützten ECN. Aber wenn auch nur ein Router in einer Verbindung ECN nicht beherrscht, dann ist ein Aufbau der Verbindung mit ECN erweiterten TCP nicht möglich. Einige Server im Internet unterstützten kein ECN [13] . Ebenso könnten bösartige Router ECN Verbindung schnell lahmlegen, was eine der schwäche von ECN. Deshalb konnte sich ECN bis heute nicht komplett im Internet ausbreiten. In kleineren Netzwerken wird es aber bereits häufig verwendet . Solange die ECN Verbindung Großteils in vertrauenswürdigen Bereichen im Internet operiert kann sie von großem Nutzen sein um eine stabile Verbindung aufzubauen.

## 3.3 Mobile IP

Mobile IP [14] wurde Mitte der 90er spezifiziert und erleichtert Mobilität von Laptops und Handys. Mobile IP ermöglicht eine interne statische IP für Anwendungen, trotz wechselnder Zugangspunkte ins Internet. Somit wäre es möglich in einem Zug oder Autos bei wechselnden Wireless Netzwerken mit einem Laptop Zugang zum Internet zu finden und Verbindungen unabhängig von der darunter liegenden Verbindung zu behalten. Gerade in der heutigen Zeit wäre dies ein großer Vorteil. Handys könnten sich in Wireless Netze einklinken und Unterhaltungen einfach weiterführen unter anderen Kostenbedingungen. Dabei

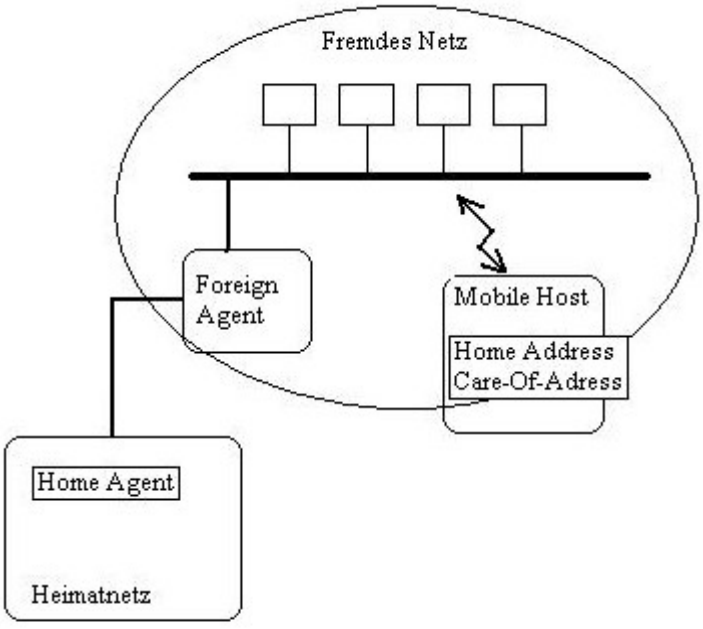



Bild 5: Mobile IP Funktionsweise

übernimmt auf Anfrage des Hosts ein Foreign Agent in einem fremden Netz die Kommunikation mit dem Home Agent im Heimnetz, der einen die statische Home Adresse verwaltet. Die Kommunikation zum Foreign Agent wird dabei mit einer temporären Care-Of-Adresse ermöglicht. Über diese Kommunikation vom Host bis zum Home Agent wird die eigentliche Kommunikation mit der statischen Adresse durchgeschleust.

Probleme gibt es aber auch mit Mobile IP. Es wäre leichter sich eine fremde IP zu hacken deren Identität anzunehmen oder dessen Verbindungen abzuhören und fortzuführen. Man müsste nur einen Foreign Agent überreden sich beim Home Agent des Opfers anzumelden, um dessen IP zu übernehmen. Eine gute und sichere Identifikation und ein sicherheitsbewusster Umgang mit Mobile IP wären eine Voraussetzung. Außerdem gibt es bei Mobile IP Probleme mit Firewall-Bedingungen, die ein Foreign Agent umgehen müsste, um bei wechselnden Netzen gleiche Bedingungen vorzufinden. Dabei werden aber Sicherheitsmechanismen von Netzen außer Kraft gesetzt, was bei den meisten Netzen nicht erwünscht ist. Wenn der Zugangspunkt in ein Netzwerk wechselt bei dem der Port eines bestehenden Dienstes aus Sicherheitsgründen gesperrt ist, so wird die Verbindung unweigerlich abbrechen. Bis heute konnte sich Mobile IP noch nicht durchsetzen im Internet. Zu viele Provider und Server bieten aus Sicherheitsgründen kein Mobile IP.

## 3.4 IPv6

Nur 4.294.967.296 eindeutige Adressen bietet IPv4. Von Anfang an war klar, dass 4 Milliarden Adressen nicht für ein weltweites Netzwerk reichen werden. Viele Lösungen wie NATs und CIDR verzögerten den Schwund von freien IPv4 Adressen. Unter anderem ermöglichen NATs, dass sich eine Organisation mit vielen Rechnern nach außen hin mit nur einer IP Adresse präsentiert. NATs trennen die Adressierung des Internets von der Adressierung in Netzwerken. Dienste werden über Portweiterleitung nach draußen geschleust. Dies bietet außerdem Sicherheitsvorteile, denn kein Host kann ohne Einstellungen am NAT Server oder Erlaubnis vom Host von außen angesprochen werden. Jedoch verletzen NATs auch die Internetphilosophie, dass jeder Host über eine eigene, eindeutige Adresse im Internet repräsentiert wird. Viele ältere Protokolle laufen deshalb bei NATs nur sehr umständlich oder gar nicht.

IPv6 löst das Problem des kleinen Adressraumes von IPv4. Mit 128 Bit kann Ipv6 ganze 340 Sextillionen Adressen darstellen. Das würde locker ausreichen jeden Quadratmillimeter Oberfläche aller Planeten im Sonnensystem samt Sonne und Monde eine eigene IP zu geben.Jedoch erfolgt die Umstellung von IPv4 auf IPv6 relativ langsam. Bereits seit fast 10 Jahren wird Hardware und Software umgestellt. Es sind erst 4% der Hardware [15][19] [20] in der Zwischenzeit umgestellt worden. Wenn man sich z.B. einen Umstellungsplan von Cisco aus dem Jahre 2002 ansieht in Bild 7, erkennt man, dass damit gerechnet wurde, dass bis 2009 der Großteil der Hardware umgestellt würde, was aber leider nicht der Fall ist. Wenigstens alle neuen Betriebssysteme und somit die Endnutzer sind inzwischen Ipv6 fähig. Jedoch wahrscheinlich wird die definitive Umstellung erst dann stattfinden wenn die IPv4 Adressen ausgehen und für die Provider und Serveranbieter keine andere Wahl besteht. Nach heutigen Statistiken müsste dies um das Jahr 2012 stattfinden.

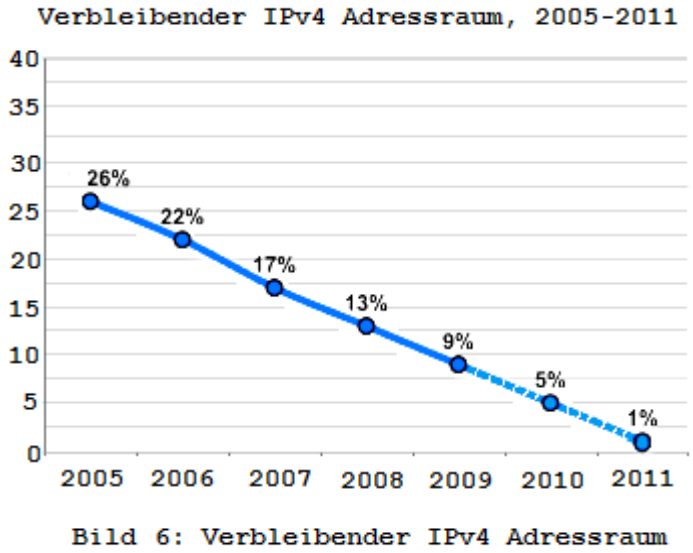



Bild 6: Verbleibender IPv4 Adressraum

Wenn man heutzutage schon Ipv6 verwenden will muss man auf Tunneldienste wie Teredo[16] und 6to4[17] zurückgreifen. Dabei werden Ipv6 Pakete in IPv4 eingepackt, an inkompatibler Hardware vorbei geschleust und an einen Server geschickt der die Pakete wieder entpackt und in das bereits aufgebaute Ipv6 Netz geschickt. Dabei werden NATs und Sicherheitsmechanismen natürlich umgangen, was aber mit IPv6 sowieso der Fall wäre. Ipv6 Adressen zu bekommen ist im Moment kein Problem, aufgrund des gewaltigen Adressraums kann man sich Bündel von Ipv6 Adressen besorgen.

Alle IPv4 Adressen können auf Ipv6 Adressen abgebildet werden, ein gängiges Schema wäre :
( IPv6 `0:0:0:0:0:ffff` c000:2800: – IPv4 192.0.2.128 )

Wie sich Ipv6 in Zukunft dann entwickelt muss man noch abwarten und wird sicherlich die interessanteste Veränderung, die im Internet im Moment, was dessen Technologie betrifft, stattfindet. Bis 2012 bleibt noch Zeit alte Hardware umzutauschen. Im schlimmsten Fall stehen uns aber große Probleme bevor wenn die IPv4 Adressen auslaufen, aber IPv6 großräumig nicht einsetzbar ist. Die meisten Privatnutzer besitzen dynamische IP die ihnen bei jedem Modem oder Routerneustart übergeben wird. Wenn die Adressen schwinden werden um die wenigen verfügbaren Adressen große Wartezeiten entstehen. Die Kosten von freien Adressen steigen. Spätestens dann würde IPv6 sich durchsetzen müssen. Auf lange Sicht führt kein Weg an Ipv6 herum.

## 4. Zukunftsaussichten

Zukünftige Änderungen an Kernsystemen im Internet wird wohl nur bei äußerster Notwendigkeit geschehen, oder über viele viele Jahre hinweg. Das Internet ist inzwischen stark etabliert und riesige Mengen an weltweiter Hardware baut auf die bestehenden Technologien auf. Viele neue Technologien bauen wiederum auf diese Kernsysteme auf und ermöglichen moderne Dienste und Anwendungen. Sämtliche Änderungen müssten viele Jahre parallel zu den momentanen Alternativen laufen können bis die Umstellung vollständig stattgefunden hat. Kleinere Änderungen könnten durch sogenannte „Killerapplikationen“ (Anwendung die einer Technologie zum Durchbruch verhilft) den Benutzern schmackhaft gemacht werden. Aber auch politische Entscheidungen und großes kommerzielles Interesse können für die Einführung von neuen Technologien helfen. Probleme im Internet gibt es viele, vor allem im Hinblick auf Sicherheit. Auch ein komplett neues weltumspannendes Netzwerk, als Konkurrenz zum Internet ist nicht ganz auszuschließen. Wir werden in Zukunft sicherlich auf Änderungen treffen, aber die werden den gleichen langsamen Weg wie Ipv6 nehmen müssen, außer das Internet würde wirklich zusammenbrechen.

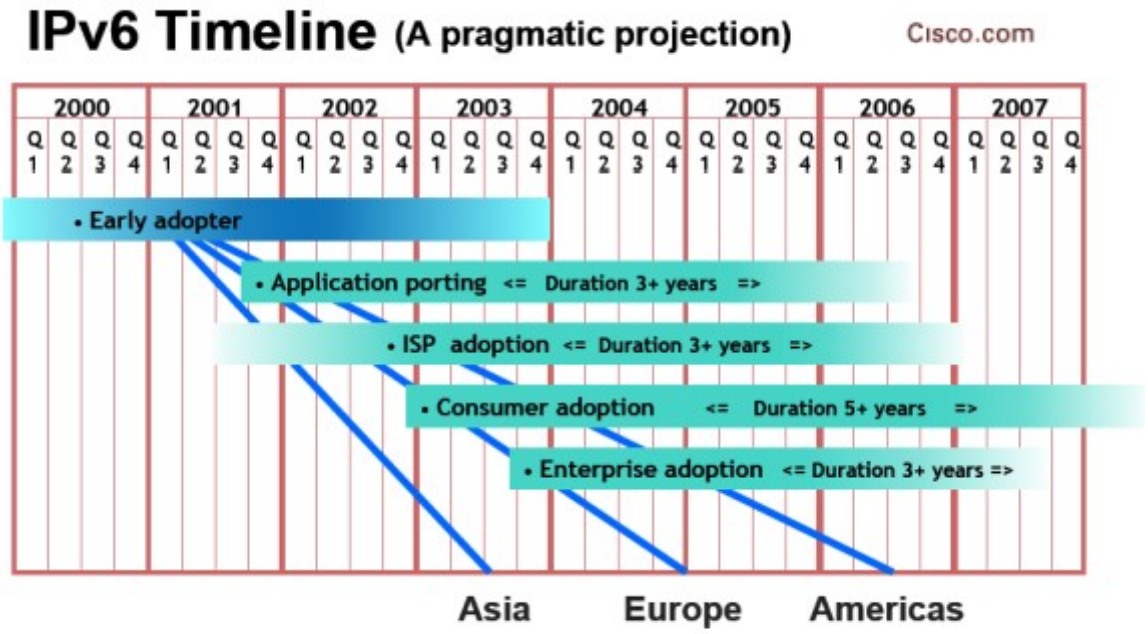



Bild 7: Cisco Umstellungsplan von 2002

## 5. Literatur


[1] M. Handly, *Why the Internet only just works*, BT Technology Journal Vol 24 No 3, Juli 2006

[2] John Day, *Patterns in Network Architecture*, Prentice Hall International, Januar 2008

[3] Steve Crocker, *Host Software* / RFC 1, April 1969

[4] S. Crocker, *Protocol Notes* / RFC 36, März 1970

[5] V. Cerf, Y. Dalal und C. Sunshune, *Specification of Internet Transmission Control Program* / RFC 675, Dezember 1974

[6] F. Mockapetris, *Domain Names Implementation and Specification* / RFC 1025, November 1987

[7] M.Allman V.PaxsonW.Stevens, *TCP Congestion Control* /TCP 2581, April 1999

[8] Chuck Semeria, *IP Aressierung: Was sie schon immer wissen wollten,* @ Com Corporation, „Juni 1996

[9] J.Mogul J.Postel, *Internet Standard Subnetting Procedure* / RFC 950, August 1985

[10] V.Fuller T.Li J.Yu K.Varadhan, *Supernetting: an Address Assigment and Aggregation Strategy* / RFC 1338, Juni 1992

[11] Y.Rekhter, T.Li, *An Architecture for IP Address Allocation with CIDR* / RFC 1528, September 1993

[12] S.Floyd D.Black, *The Addition of Explicit Congestion Notification (ECN) to IP* / RFC 3168, September 2001

[13] D.Hargreaves G. Sittampalam, *ECN Page*, März 2005 http://urchin.earth.li/ecn/

[14] C. Perkins, *IP Mobility Support for Ipv4* / RFC 3344, August 2002

[15] Pingdom, *A crisis in the making Only 4% of the Internet supports Ipv6,* März 2009 http://royal.pingdom.com

[16] J.Endres,R. Kaps, *Teredo bohrt IPv6-Tunnel durch Firewalls*, Heise Netz, März 2009

[17] Hubert Feyrer, *Rapid Deployment von IPv6 mit 6to4 ,* September 2001

[18] M. Małowidzki, *ECN is fine – but will it be used?,*Military Communication Institute,Poland

[19] Derek Morr, Why 2008 Was a Milestone Year for Ipv6, Januar 2009

http://www.circleid.com

[20] Statistische Daten von CIDR:

http://www.cidr-report.org

# Überblick und Vergleich von Sensorknotentechnologien

Korbinian Michael Mögele
Betreuerin: Corinna Schmitt
Seminar Sensorknoten: Betrieb, Netze & Anwendungen SS 2009
Lehrstuhl Netzarchitekturen und Netzdienste, Lehrstuhl Betriebssysteme und Systemarchitektur
Fakultät für Informatik
Technische Universität München
Email: moegele@in.tum.de

## Kurzfassung

In dieser Arbeit soll ein Überblick über Sensornetze und die in ihnen verwendeten Technologien gegeben werden. Es existieren verschiedene Ansätze, die jeweils ihrer Domäne entsprechend auf unterschiedliche Zielsetzungen ausgerichtet sind. Smart-Its zielen auf eine dezentralisierte Berechnung ab, bei der das physikalische Phänomen im Vordergrund steht. Verschiedene Sensoren sind nach dem "Plug-and-Play" Prinzip an ein Sensorboard anschließbar. Bei ScatterWeb ist eine spezielle Optimierung bis zum batteriefreien Betrieb erfolgt. Diese Sensorknoten können Energie mittels verschiedener Module aus der Umwelt gewinnen. Darüber hinaus steht vor allem die Selbstkonfiguration und Wartungsfreiheit im Vordergrund. Die Mica-Technologie arbeitet mit einem extrem auf Energieeffizienz ausgerichteten und minimalistischen Betriebssystem, TinyOS. Entsprechend ist hier die Energie- und Zeiteffizienz wesentliche Optimierungs-grundlage. Die Vorstellung und der Vergleich dieser drei unterschiedlichen Technologien im Bereich Sensornetze soll Bestandteil dieser Ausarbeitung sein. Dabei gibt es keine allgemein beste Technologie, die verwendete richtet sich vielmehr nach den entsprechenden Anforderungen, Leistungskriterien und Zielsetzungen des entsprechenden Anwendungsszenarios.

## Schlüsselworte

Sensornetze, Sensorknoten, Smart-Its, ScatterWeb, Energie-bewusstes-Routing, Mica-Technologie, TinyOS, Technologie-vergleich

## 1. Einführung

Verteilte Sensornetze bestehen meist aus einigen Hundert, in großen Szenarien sogar aus bis zu einigen Millionen Sensorknoten [1]. Die Spezifikationen derartiger Sensorknoten unterscheiden sich drastisch von gängigen Desktop/Server Konstellationen. Prämissen waren und sind, eine gegebene Funktionalität in immer kleinere und billigere Bausteine zu integrieren [2]. Diese Bausteine sollen zudem noch eine möglichst effiziente und geringe Energiebilanz aufweisen. Desweiteren wird dem Ziel nachgegangen, komplette Systeme, das heißt Betriebssystem, Kommunikation und Sensoren, auf einem Chip zusammenzufassen. Aus diesen zwei Ansätzen folgt direkt, eine Kommunikation zu ermöglichen, die auf dem Chip direkt integriert sein soll und zudem sehr wenig Energie verbraucht. Diese Entwicklungen haben dazu geführt, dass verteilte Sensornetze in dem Maße, wie sie heute bereits dem Stand der Technologie entsprechen, überhaupt erst möglich wurden.

Die verschiedenen Technologien von Sensorknoten können dabei nach diversen Kriterien eingeteilt und bewertet werden. Besondere Bedeutung bei verteilten Sensornetzen kommt der Kommunikation zu. Diese kann mittels verschiedenster Technologien wie Infrarot, optisch, mittels Radio-Frequenz oder vieler anderer erfolgen [3]. Die verschiedenen Umwelteinflüsse beziehungsweise Phänomene, die mit den Sensorknoten gemessen, registriert und protokolliert werden können, unterscheiden sich je nach verwendeten Sensoren. Es existieren beispielsweise Sensoren für die Messung von Licht, Hitze, Bewegung, chemischen Prozessen und andere [3]. Darüber hinaus gibt es immer wieder Weiter- und Neuentwicklungen für entsprechende Sensoren. Nötig machen dies entweder neue qualitative Anforderungen, die momentane Lösungen nicht erfüllen können oder neu erschlossene Anwendungsszenarien. Für Berkeley Motes wurde beispielsweise an der Universität von Kalifornien der Illumimote entwickelt, ein Lichtsensor, dessen Leistung mit einem kommerziellen Belichtungsmesser vergleichbar ist [4].

Verteilte Sensornetze und damit auch die Sensorknoten müssen dabei äußerst adaptiv sein und sich mit vielen verschiedenen Umwelteinflüssen arrangieren können. So ist es möglich, Sensorknoten auch in sogenannten feindlichen Gebieten einzusetzen [1]. Diese Gebiete können entweder im militärischen Bereich durch feindliche Truppen oder aber durch natürliche Bedingungen, wie Radio-Aktivität, gekennzeichnet sein. Es entstehen eine Vielzahl an unterschiedlichen Einsatzmöglichkeiten und somit auch Anforderungen an die Sensorknoten und –netze. Verschiedene Technologien haben sich entsprechend spezialisiert und es ergeben sich, auf eine Umweltsituation bezogen, entsprechend mehr und weniger geeignete technische Umsetzungen. Das MediaCup Projekt [5] war ein in einen Kaffeebecher eingebettetes Sensornetz, das selbstständig seinen momentanen Benutzerkontext berechnet (z. B.: Kaffeebecher leer oder voll). Dieser kann dann an interessierte Anwendungen in der Umgebung übertragen werden. Hauptsächlich werden Sensornetze zum Beoabchten oder Erkennen von Phänomenen eingesetzt [6]. Willig [7] beschreibt ein Sensornetz, das das Erkennen und Melden von Feuern in einem Waldgebiet ermöglicht.


*Abbildung 1: MediaCup [5]*

In dieser Arbeit sollen drei der bekanntesten Technologien im Bereich Sensornetze untersucht werden. Das Smart-It Projekt wurde 2000 von der Europäischen Union unter der Forschungsinitiative “The Disapperaing Computer” initiiert [6]. ScatterWeb ist ein mittlerweile kommerzielles Projekt, das an der Freien Universität Berlin entwickelt wurde [8]. Eines der wohl bekanntesten Beispiele für Sensornetze sind die an der Universität von Kalifornien entwickelten Berkeley-Motes [8], von denen die Mica-Motes eine Untergruppe sind. In der Literatur werden diese auch oft nur als *motes* bezeichnet. Diese drei Technologien sollen hinsichtlich ihrer Hardwareaussattung, der jeweils verwendeten Kommunikation und deren Leistungsmerkmalen betrachtet werden. Desweiteren soll auf limitierende Faktoren, Optimierungsmöglichkeiten und auf entsprechende Einsatzgebiete eingegangen werden.

Im Folgenden wird zunächst in Kapitel zwei das Smart-Its Projekt vorgestellt und auf beschriebene Merkmale eingegangen. Darauf folgt das ScatterWeb-Projekt der Freien Universität Berlin in Kapitel drei. Die Berkeley-Technologie wird in Kapitel vier untersucht, gefolgt von einem Vergleich aller drei Technologien hinsichtlich Ähnlichkeiten und Unterschiede in Kapitel fünf.

## 2. Smart-Its

Das Smart-Its Projekt ging aus einer Forschungsinitiative der Europäischen Union (EU) hervor. Aufgrund dieser Tatsache gibt es unterschiedliche Entwicklungen an verschiedenen Forschungseinrichtungen in ganz Europa. Beispielsweise entwickelte die Universität von Lancester die *Lancaster DIY Smart-Its* [9] oder die Universität Karlsruhe die *TecO Particle Smart-Its* [9]. Von denen die erstgenannten speziell für die einfache Hard- und Software Anpassung entwickelt wurden und die zweitgenannten auf eine miniaturistische Größe und eine optimale Energie- und Kommunikationseffizienz abzielen [9].

### 2.1 Konzept

Der Fokus liegt auf den beteiligten physikalischen Elementen (z.B. den Sensoren) nicht auf den Berechnungen. Diese sollen gemäß der Initiative der EU mehr in den Hintergrund rücken. Ein wichtiges Konzept bei den Smart-Its ist das in der Informationstechnologie unter *Plug-and-Play* bekannte Prinzip: Sensoren und/oder Aktoren sollen nachträglich und ohne Konfiguration zu bestehenden Smart-Its hinzugefügt oder ausgetauscht werden können. Darüber hinaus sollen die einzelnen Sensorknoten einen hohen Grad an Autonomie aufweisen und sich zum größten Teil selbst und autark steuern. Diese Autonomie soll sich in einer langen Lebenszeit niederschlagen, die sich dadurch auszeichnet, dass ein externer Benutzereingriff (z.B. für eine Neukonfiguration) selten bis gar nicht nötig ist. Die einzelnen Sensorknoten sollen somit klein und leicht sein, sowie eine hohe Energieeffizienz aufweisen.

### 2.2 Hardware und Systemsoftware

Es exisitieren zwei Hardware Module, eines für die Kommunikation (*core board*) und eines für die Sensoren, also die Interaktion mit der Umwelt [6]. Diese beiden Module werden durch einen I2C Datenbus verbunden, darüber hinaus existiert ein Energiebus (die Stromversorgung, die durch verschiedene Batteriearten erfolgt, ist auf dem Core Board platziert). Die Struktur und das Zusammenwirken dieser beiden Module ist in Abbildung 2 zu sehen. Da sowohl Core Board als auch das Sensorboard über einen eigenen Prozessor verfügen (20 MHz RISC Arizona Microchip PIC Prozessor), können sensorspezifische Berechnungen direkt auf dem Sensor Board durchgeführt werden und belasten nicht das Core Board. Darüber hinaus werden so mehrere Sensoren auf einem Sensor Board möglich. Die spezielle Bus-Struktur erlaubt es außerdem, nicht nur ein Sensor Board zu installieren, sondern mehrere. Auf dem Core Board ist für die drahtlose Kommunikation ein Empfänger installiert (RFM Empfänger Modul TR 1001 mit 868,35 MHz).

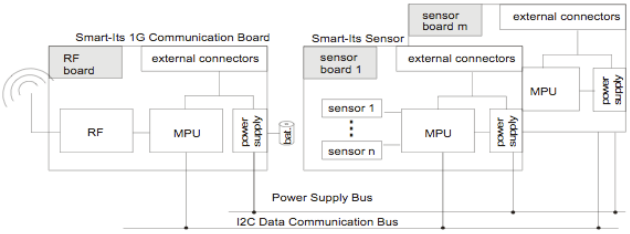



Abbildung 2: Die zwei Smart-IT Module mit Power-/Databus [6]

Die Smart-Its verfügen über ein eigenes Betriebssystem (OS), das aus drei großen Bibliotheken aufgebaut ist: Der *Sensor/Actuator Lib*, für die Unterstützung von Sensoren und Aktoren, die *Communication Lib*, für die Kommunikation mit anderen Sensorknoten und die *OS Core Lib*, die Funktionen wie Timer und Semaphore zur Verfügung stellt. Da der Fokus auf Energieeffizienz liegt, ermöglicht das OS das direkte An- und Abschalten jeglicher Komponenten. Dies kann entweder system-kontrolliert erfolgen, oder anwendungs-kontrolliert.

### 2.3 Kommunikation

Die Smart-Its sind so konzipiert, dass eine Kommunikation in zufällig arrangierten Sensornetzen von Smart-Its problemlos möglich ist. Das bedeutet, es ist keine Konfiguration von außen nötig, die Smart-Its konfigurieren sich selbst. Die Smart-Its kommunizieren dabei mittels einer Radiofrequenz über kurze Strecken. Das Hauptaugenmerk liegt hier auf der schnellen Netzwerkerforschung und dem Austausch der verschiedenen Sensorkontexte. Die Datenübertraung selbst basiert dabei auf einem zustandslosen *peer-to-peer* Protokoll. Um dabei Kollisionen von Paketen der verschiedenen Smart-Its beim Senden/Empfangen zu vermeiden, wird auf der physikalischen Ebene ein *time division multiplex collision avoidance* (TDMCA) Protokoll eingesetzt. Dabei werden die Pakete im *next-hop* Verfahren von Smart-IT zu Smart-IT an eine zentrale Stelle weitergeleitet, die die Informationen dann weiterverarbeitet. Die einzelnen Sensorknoten sind also nicht für eine Verarbeitung von Informationen verantwortlich, die sie selbst erhalten, sie fungieren im Sensornetz vielmehr als viele kleine Router. Das Kommunikationsmodul der Smart-Its ist aufgrund seiner hohen Anforderungen der Energieeffizienz, Ziel vieler Weiterentwicklungen beziehungsweise neuer technischer Entwicklungen. So beschreiben Oliver Kasten und Marc Langheinrich den Einsatz von *Bluetooth* Modulen [10].

### 2.4 Limitierende Faktoren und Optimierungsmöglichkeiten

Aufgrund der technischen Möglichkeiten bestehen bei den Smart-Its Einschränkungen, was umsetzbar ist und was nicht. Dabei ist zum einen die ständige Verringerung der Größe erwähnenswert, die momentan nur bis zu einem bestimmten Grad möglich ist. Zum anderen spielt die Energie-Effizienz eine große Rolle. Aus diesem Grund ist fast jede Komponente eines derartigen Sensorknotens Ziel von Optimierung hinsichtlich Größe und Energieeinsparung. In den letzten Jahren haben sich immer wieder effizientere Möglichkeiten ergeben [10]. Was die Skalierbarkeit bezüglich der Größe des entsprechenden Sensornetzes betrifft, so ist das Smart-Its Projekt so ausgelegt eine uneingeschränkte

Anzahl Sensorknoten zu ermöglichen [6]. Smart-Its sind jedoch auf einen Batteriebetrieb angewiesen und diese müssen zwangsläufig irgendwann ausgetauscht werden.

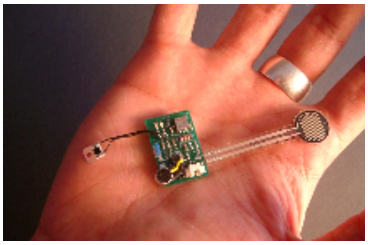


Abbildung 3: Ein Smart-IT Sensorknoten [11]

## 2.5 Einsatzgebiete

Smart-Its können prinzipiell so angepasst werden, dass sie auf eine Vielzahl von Einsatzmöglichkeiten anwendbar sind. Durch das nachträgliche Hinzufügen und Austauschen von Sensoren/Aktoren sind sie zudem an entsprechend wechselnde Umweltbedingungen anpassbar. Smart-Its können zudem theorethisch an sehr unterschiedliche Alltagsgegenstände angebracht werden, um diese mit Sensoren/Aktoren und damit einer Art Intelligenz auszustatten [11].

In der Vergangenheit wurde beispielsweise mithilfe von Smart-Its das Zusammenbauen von Möbeln beobachtet und erleichtert [12]. Indem an lose Möbelteile Smart-IT Sensoren, wie in Abbildung 3 zu sehen, angeheftet wurden, konnte der Fortschritt beim Zusammenbau beobachtet werden. Eine Rückmeldung beziehungsweise Informationen konnten infolgedessen an den Benutzer weitergegeben werden.

# 3. ScatterWeb

ScatterWeb ist ein an der Freien Universität Berlin entwickeltes Konzept eines Sensornetzes [8]. Zunächst ein reines Forschungsprojekt ging daraus 2005 die kommerzielle ScatterWeb GmbH hervor [13].

## 3.1 Konzept

Neben den an Sensornetze und Sensorknoten gestellten Anforderungen, lag der Fokus bei der Entwicklung von ScatterWeb vor allem auf der Energieeffizienz und dem sogenannten *Energy Scavenging* [2]. *Energy Scavenging* bedeutet, die für die Sensorknoten nötige Energie nicht allein durch Batterien bereit zu stellen, sondern erneuerbare Energien aus der Umwelt zu nutzen [14]. Diese Energiegewinnung kann durch Solarmodule, aus Vibrationen oder Gravitation erfolgen. ScatterWeb hat damit auch den Begriff des *energy aware routing*, wie in [15] beschrieben, geprägt. Thiemo Voigt geht in [16] besonders auf *solar aware routing* ein, dessen Algorithmus lässt sich aber problemlos auf andere Energiegewinnungen übertragen. Das bedeutet, Pakete werden von Sensorknoten zu Sensorknoten geroutet, und zwar präferenziert über Knoten, die über eine Energiegewinnung aus erneuerbaren Energien verfügen. So wird ein möglichst hohes Maß an nicht erneuerbarer Energie (bei den Sensorknoten mit Batteriebetrieb) gespart und die Lebensdauer der Sensorknoten deutlich verlängert. Jochen Schiller et al haben in [2] gezeigt, dass mitunter Energieeinsparungen in Sensornetzen gegenüber einem herkömmlichen Routing von bis zu 20% möglich sind. Wichtiges Designkonzept bei ScatterWeb ist überdies sich vollständig selbst konfigurierende Knoten und ein daraus resultierendes möglichst wartungsfreies Sensornetz. Diese Wartungsfreiheit äußert sich durch die Möglichkeit, eine Neuprogrammierung (z.B. ein Update der Systemsoftware) der Sensorknoten aus der Ferne durchzuführen, die von Sensorknoten zu Sensorknoten weiterpropagiert wird.

Abbildung 4: Embedded Sensor Board (ESB) [2]

## 3.2 Hardware und Kommunikation

ScatterWeb verfügt über zwei verschiedene Bausteine, die ein entsprechendes Sensornetz auszeichnen: *Embedded Sensor Board (ESB)* und *Embedded Web Server (ESW)*. ESBs (zu sehen in Abbildung 4) stellen die Sensorknoten dar, verfügen über eine vorher festgelegte Anzahl an Sensoren. Sie propagieren mittels eines *peer-to-peer* Protokolls ihre gesammelten Informationen an benachbarte ESBs oder EWSs weiter. Ein ESB verfügt über einen Microcontroller von Texas Industries, einen MSP430. Zunächst war ein relativ einfacher Radioempfänger installiert, ein auf 868MHz laufender RFMTR1001. Dessen Reichweite genügte allerdings bald nicht mehr den Anforderungen (ca. 300m auf freiem Gelände und ca. 30m in Gebäuden), sodass an dieser Stelle bald optimiert und der neuere Radioempfänger von Chipcon, der CC1021, eingesetzt wurde. Dieser verfügt über eine 10mal größere Reichweite. Die Sensorknoten verbrauchen im Betrieb mit allen Sensoren gerade einmal 12mA, beim Übertragen von Daten weniger als 8mA und wenn der *deep-sleep* Modus genutzt wird ca. 8µA [2]. Darüber hinaus existiert ein sogenannter *secure-shutdown* Modus, der besonders bei Sensorknoten, die ihre Energie aus der Umwelt beziehen von Bedeutung ist. Dabei wird der momentane Energievorrat des Sensorknotens ständig beoabachtet, eine Weiterleitung von Paketen erfolgt nur bei ausreichender Energie. Geht das Energieniveau in einen kritischen Zustand über, fährt sich der Sensorknoten selbstständig in einen sicheren Zustand runter (deep-sleep). Dadurch werden inkonsistente Zustände und Fehlverhalten der Sensorknoten verhindert.

Die EWSs sind spezielle Sensorknoten, die eine Kommunikation nach und von außerhalb des Sensornetzes ermöglichen. Sie verhalten sich im Prinzip wie *backbone* Router, mehrere EWSs zusammen bilden ein *backbone* zum Sensornetz. Sie sind an externe Netzwerke, das Internet oder ähnliches angebunden und können über verschiedene Technologien angesprochen werden (z.B. GPRS, Bluetooth, Ethernet usw.). Die Kommunikation von außen mit den EWSs erfolgt dabei meist über das *transmission control protocol* (TCP), da auf den EWSs ein simpler Webserver implementiert ist.

Bei der Kommunikation spielt die Selbstkonfiguration der Sensorknoten eine Rolle, denn diese erkunden die Netztopologie automatisch. Bevorzugte Routen (siehe *energy aware routing*) werden gespeichert und wiederverwendet. Dabei kann ein Packet mit Flags ausgestattet werden, die entweder eine schnell, oder

eine möglichst energieeffiziente Zustellung fordern [16]. Die Paketzustellung erfolgt im *next-hop* Verfahren nach einem *peer-to-peer* Protokoll.

Das Zusammenspiel von ESBs und EWSs ist in Abbildung 5 illustriert.

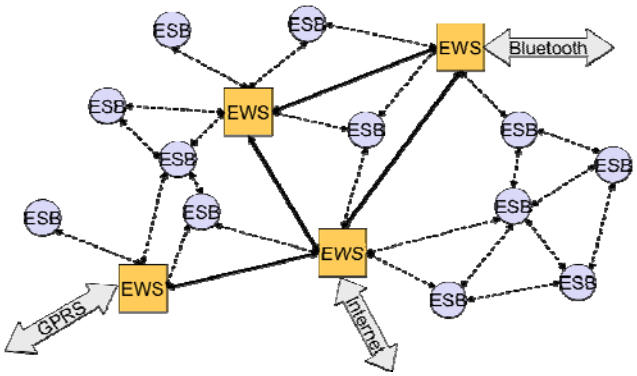



Abbildung 5: Ein ScatterWeb Sensornetzwerk [2]

## 3.3 Limitierende Faktoren und Optimierungsmöglichkeiten

Das relativ große Sensor Board mit allen Sensoren wird durch einen anderen Typ von ESBs ergänzt, dessen Dimensionen deutlich geringer sind, dem *Embedded Chip Radio* (ca. 7x4x6cm) [8]. Bei diesem ist außerdem ein modulares Hinzufügen von Sensoren möglich im Gegensatz zum fest installierten *ESB*. Der zunächst verwendete Radioempfänger wurde durch eine Alternative mit höherer Leistung ersetzt. Für die Skalierbarkeit von ScatterWeb Netzwerken gilt, dass diese in nahezu unbegrentzer Menge erweiterbar sind[2]. Für das *energy aware routing* ist festzustellen, dass bei sehr großen Netzen teilweise nur Einsparungen von 6 % möglich sind [16]. Hier besteht sicherlich noch die Möglichkeit, diese Routing-Protokolle entsprechend zu optimieren.

## 3.4 Einsatzgebiete

ScatterWeb Netzwerke lassen sich sehr vielseitig einsetzen. Besonders das Nutzen von erneuerbaren Energien macht diese Sensorknoten vor allem für unerreichbare und wartungfreie Szenarien attraktiv. Die ScatterWeb GmbH setzt diese Netzwerke u.a. für Energie-Management ein [13]. Es werden relevante Informationen gesammelt, Energieverbraucher gesteuert und in Abrechnungssysteme angebunden. Ein weiteres großes Anwendungsszenario ist die Bauwerksüberwachung und damit verbunden die Überprüfung der Bauwerkszustände und Wartungskontrolle. Desweiteren existieren Geoinformations- und Kontrollsysteme, wie in [8] für ein Offshore Temperaturmesssystem im Meer beschrieben.

# 4. Berkeley-Technologie

Die Berkeley-Motes, die in der Literatur auch lediglich als *motes* bezeichnet werden, wurden an der Universität von Kalifornien entwickelt. Nach [8] sind die *Motes* eines der bekanntesten Beispiele für Sensornetze. Diese Technologie setzt sich aus verschiedenen Knoten (Telos, Mica, Iris) zusammen, die mit unterschiedlicher Hardware ausgestattet sind. In dieser Arbeit wird nur auf die Mica-Technologie eingegangen. Die vom Hersteller Crossbow produzierten Mica-Motes sind auf extrem geringe Hardwareanforderungen getrimmt, was mitunter auch durch das verwendete Betriebssystem TinyOS (und dessen geringen Speicherbedarf) erreicht wird [3, 17].

## 4.1 Konzept

Bei Mica steht im Vordergrund, eine drahtlose Verbindung und Verfügbarkeit billiger und tief integrierter Geräte zu erreichen. Tiefe Integration kann in diesem Zusammenhang beispielsweise den Einbau von Mica-Knoten in unerreichbare Komponenten einer Brückenkonstruktion bedeuten. Die gesammelten Informationen sollen dann von einer großen Bandbreite von Anwendungen genutzt werden können [17]. Eine der Prämissen von Mica-Motes ist das Sammeln von Sensorendaten in großem Stil und großen Netzwerken. Im Gegensatz zu ScatterWeb, bei dem es besonders um die Energieeffizienz geht, kommt dem Ziel der Zeiteffizienz bei Mica ebenso große Bedeutung zu wie der Energieeffizienz. Um eine zeitsynchrone Übertragung zu ermöglichen kommen zusätzliche Bausteine auf den Mica-Motes zum Einsatz. Dadurch treten die Schlüsselfähigkeiten des System zum Vorschein: sehr billiges Starten des Gerätes, Zeitsynchronisierung, Energie bewusstes Routing (ähnlich ScatterWeb) und Lokalisierung von Mica-Motes [17, 18].

## 4.2 Hardware und Systemsoftware

In Abbildung 6 sind die Kernelemente der Mica Architektur zu sehen.

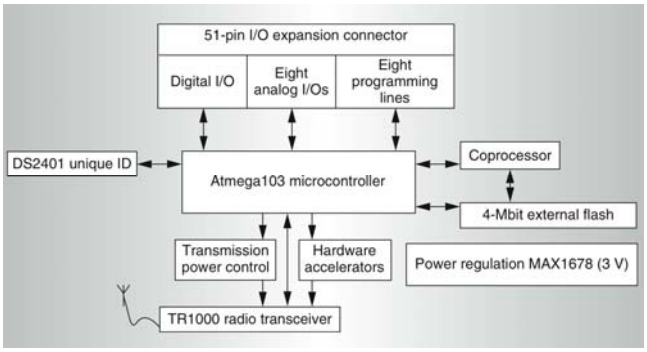



Abbildung 6: Mica Architektur[17]

Herzstück bildet der Atmega 103 Mikrokontroller, der mit 4MHz läuft. Er lieftert ungefähr vier Millionen Befehle pro Sekunde und ist mit 128-Kbyte Flash-Memory, 4-Kbyte statischem Arbeitsspeicher und neben anderem mit 48 I/O Verbindungen ausgestattet [17]. Einzigartig in der Technologie ist auch das Bauteil Maxim DS2401, ein ROM Bauteil, das eine eindeutige Serienummer zur Identifikation des Mica-Mote im Sensornetzwerk liefert. Der Koprozessor dient dazu, eine Aktualisierung des Betriebsssytems auf dem Sensorknoten vorzunehmen. Der Radioempfänger besteht aus dem RF Monolithics TR1000 und weiteren Modulen, die zum Beispiel für die Steuerung zuständig sind. So ist eine Softwaresteuerung u.a. der Sendeleistung des Radioempfängers möglich, um so die Reichweite von wenigen Zentimetern auf über hundert Meter einzustellen. Für den Anschluss von Sensoren existiert ein 51-pin I/O Anschluss. An diesen können beliebige Bauteile mit Sensoren angeschlossen werden, Mica-Motes entsprechen also auch einem modularen Sensorkonzept.

Das Betriebssystem der Mica-Motes stellt das TinyOS dar, das ebenfalls an der Universität von Kalifornien, Berkeley, entwickelt wurde. Es handelt sich um ein Multithreading Betriebssystem, was dazu führt, dass verschiedene Prozesse nicht blockiert werden können und scheinbar simultan ausgeführt werden (bspw. die Verarbeitung von Sensordaten mit einem plötzlich einsetzenden Empfang eines Paketes). TinyOS zeichnet sich darüber hinaus durch seine extrem geringen

Speicherplatzanforderungen aus [3]. Die Energieeffizienz wird hauptsächlich auch durch das Betriebssystem erreicht, das jegliche Hardwarekomponenten ausschaltet, wann immer dies möglich und effizient ist. Dazu existieren drei verschiedene Schlaf-Modi in TinyOS: *idle*, *power down* und *power save*.

Von der Mica-Technologie exisitieren grundsätzlich zwei verschiedene Typen: die Mica-Motes und die MicaDot-Motes [Abb. 7]. Erstere haben ungefähr eine Größe von 5x3x4cm, letztere sind erheblich kleiner, da sie mit einer extrem begrenzten Anzahl Sensoren ausgestattet sind und entsprechend kompaktere Bauteile verwendet werden. Sie haben lediglich einen Durchmesser von ca. 25mm.

## 4.3 Kommunikation

Bei der Kommunikation kommt wie bei SmartITs und ScatterWeb ein *peer-to-peer* Protokoll zum Einsatz. Dieses gewährleistet eine hohe Erreichbarkeit bei gleichzeitig geringerem Verbrauch für das Senden von Paketen (geringere Sendeleistung des Radioempfängers). Die Sendeleistung des Radioempfängers kann stufenlos geregelt werden, sodass der Energieverbrauch an die speziellen variablen Entfernungen zwischen den Sensorknoten angepasst werden kann. Ein entsprechendes Protokoll, das diese Faktoren berücksichtigt wird in *The Mote Connectivity Protocol* [19] beschrieben. Durch einen Buffer, der dem Mikrokontroller und dem Radio Empfänger zwischengestaltet ist, können extrem kurze Sendezeiten erreicht werden. Eine flexible und zuverlässige Implementierung dieses Buffers wird in [20] beschrieben. Sensorknoten, die sich momentan in einem der Schlafmodi befinden, können zudem über ein speziell empfangenes Signal aus dem Sensornetzwerk wieder erweckt werden [17]. Dadurch wird die Energieeffizienz der Knoten weiter optimiert, da ein aktives Kontrollieren, ob gerade eine neue Nachricht aus dem Sensornetzwerk eintrifft, entfällt. Der Mica-Mote erwacht nur zum Leben, wenn dies wirklich notwendig ist.

Abbildung 7: Mica-Mote und münzgroße MicaDot-Mote [8]

## 4.4 Limitierende Faktoren und Optimierungsmöglichkeiten

Für Mica-Motes ergeben sich keine weiteren relevanten Ansatzpunkte, wie bei den anderen beiden Technologien, Smart-Its und ScatterWeb. Grenzen und Möglichkeiten lassen sich an entsprechenden Merkmalen klassifizieren. Erwähnenswert ist die Arbeit von G. Anstasi u.a. [21]. Er untersuchte den Einfluss von Umweltbedingungen, wie Regen oder Nebel, auf die Kommunikationsleistung in Sensornetzen. Drastische Einbrüche der Reichweite ergaben sich beispielsweise bei Regen oder der zu bodennahen Positionierung von Mica-Motes.

## 4.5 Einsatzgebiete

Nach Jason L. Hill in [17] lassen sich Anwendungsszenarien im wesentlichen in zwei Kategorien einteilen. Zum einen Anwendungen, die sich durch eine kurze Betriebszeit im Vergleich zur Ruhezeit, durch eine geringe Datenrate, eine große Verzögerung bei der Datenübertragung, eine mitunter statische Netztopologie und eine lange zu erwartende Lebensdauer auszeichnen. In diesen Netzwerken wird die Qualität hauptsächlich an der erreichten Lebensdauer gemessen. Die andere Kategorie wird durch eine genau zu erreichende Verzögerung bei der Datenübertragung und eine hohe Datenübertragungsrate evaluiert. Hier sind die Mica-Motes zudem durch eine hohe Mobilität ausgezeichnet.

In *Motes Sensor Networks in Dynamic Scenarios* [22] wird eine experimentielle Studie eines Sensornetzwerkes der zweiten Kategorie untersucht. Sich bewegende Objekte mit Sensoren versuchen bei verschiedenen Geschwindigkeiten Kontakt mit fest installierten Sensoren aufzunehmen und entsprechend Daten auszutauschen. In der Studie wird ein Anwendungsszenario von in oder an der Straße installierten Sensoren und vorbeifahrenden Bussen gezeichnet.

# 5. Vergleich der Technologien

Bei allen drei Technologien, Smart-Its, ScatterWeb und Mica-Motes, lässt sich die gleiche Zielsetzung bezüglich Energieeffizienz, Größe des Sensorknotens (dieser soll möglichst klein sein) und Anpassungsfähigkeit an unterschiedliche Umweltbedingungen erkennen. Die Umsetzung und das jeweilig vorranige Ziel unterscheidet sich jedoch in der jeweils betracheten Technologie.

## 5.1 Energie

Smart-Its und Mica-Motes sind auf einen Batteriebetrieb angewiesen. Zwar werden unterschiedliche Batterietypen für entsprechende Anforderungen unterstützt, aber die Abhängigkeit von einer festen Energiequelle bleibt. Diese Abhängigkeit wirkt sich negativ auf die Lebensdauer der entsprechenden Sensorknoten aus. Zwar gibt Jason L. Hill in [17] eine Lebensdauer von ungefähr 10 Jahren bei Betrieb mit zwei AA-Batterien an. Dieser Wert ist jedoch eher theorethischer Natur, da die Haltbarkeit der Batterien diese Lebensdauer überhaupt nicht ermöglicht. Smart-Its und Mica-Motes unterliegen demzufolge früher oder später der Notwendigkeit einer Wartung. Im Gegensatz dazu besteht bei ScatterWeb die Möglichkeit, neben einem Batteriebetrieb, auch Module einzusetzen, die ihre Energie aus der Umwelt gewinnen [14]. Da entsprechende natürliche Ressourcen in manchen Gebieten vielleicht nicht dauerhaft vorhanden sind, sind diese Sensorknoten unter Umständen nicht dauerhaft einsatzbereit. Jedoch entfällt auf lange Sicht gesehen die bei den anderen Technologien notwendige Wartung. Die Sensorknoten sind vollständig autark. Zudem unterscheiden sich die Sensorknoten der verschiedenen Technologien grundsätzlich in ihrem jeweiligen Energieverbauch. Dies liegt vor allem in verschiedenen Bauteilen aber auch den unterschiedlich unterstützten Schlaf-Modi in den speziellen Betriebssystemen begründet.

## 5.2 Betriebssystem

Die Mica-Motes benutzen das externe Projekt TinyOS als Betriebssystem. Damit wird eine Modularisierung erreicht, die die Vorteile eines externen Betriebssystems nutzen kann. Hinzu kommt, dass TinyOS ein OpenSource Projekt ist, was weitere Vorteile mit sich bringt. Im Gegensatz dazu setzen Smart-Its und ScatterWeb auf eigene Implementierungen eines Betriebssystems. Inwiefern dies einen Vor- oder Nachteil darstellt bleibt schwierig zu beurteilen. Eigene Implementierungen können auf Systembesonderheiten besser eingehen und diese zu ihrem Vorteil nutzen. Andererseits sind sie durchaus fehleranfälliger und bedürfen größerer Sorgfalt, als dies bei einem funktionierenden

und sicheren OS wie TinyOS der Fall ist. Das Betriebssystem kann meist *remote*, d.h. aus der Ferne, nach der Installation der Sensorknoten aktualisiert werden. Bei Smart-Its beispielsweise wird dieses Verfahren als *AirProg* bezeichnet.

## 5.3 Flexibilität und Sensormodularität

Einige der Technologien sind in Bezug auf die Umweltbedingungen durchaus als flexibler einzustufen als andere. So ist es bei den Smart-Its nachträglich problemlos möglich verschiedenste Sensor-Boards hinzuzufügen, bestehende auszutauschen oder sogar mehrere Sensor-Boards gleichzeitig zu betreiben. Auf den Sensor-Boards befindet sich ein entsprechender Mikrokontroller, der für die Sensoren verantwortlich zeichnet. Dadurch ist es auch möglich Sensor-Boards verschiedenster Hersteller zu kombinieren, Konflikte sind bei richtiger Implementierung der Interfaces ausgeschlossen. Bei ScatterWeb und Mica-Motes besteht zwar auch ein hoher Grad an Flexibilität, bei ScatterWeb muss jedoch die entsprechende Kombination von Sensoren vorher festgelegt werden. Bei Mica-Motes ist ein Austausch des Sensor-Boards auch nach Installation noch möglich, jedoch kann eine Variation nur auf einem Sensor-Board stattfinden. Mehrere Sensor-Boards können nicht gleichzeitig genutzt werden, vielmehr müssen die benötigten Sensoren auf einem Board zusammengefasst werden.

| Hardware: | Smart-Its | ScatterWeb | Berkeley (Mica2) |
|---|---|---|---|
| RAM | 8k x 14 Wort Flash 368 Byte Daten | 5 K-Byte | 4 K-Byte Static |
| ROM | 256 K-Byte | 55 K-Byte Flash | 128 K-Byte Flash |
| Energieressourcen | Batterie | Batterie / erneuerbare Energien | Batterie |
| Funkart | Radiowelle, Bluetooth u.a. | Radiowelle (RF Monolithics TR1001) u.a. | Radiowelle (Chipcon) u.a. |
| Prozessor | 20 MHz RISC Arizona Microchip PIC Processor | MSP430 (Texas Instruments) | Atmega103 microcontroller |
| Größe in cm | 5x5x1 (core board) | 3x4x1 | 3,17x3,17 bis zu 2,5x2,5x0,5 Münze |

Abbildung 8: Hardwarevergleich der Technologien

## 5.4 Kommunikation

Bei der Kommunikation in den Sensornetzen bestehen große Analogien. Alle Technologien bauen ihr Kommunikationsprotokoll auf dem *peer-to-peer* Protokoll auf. Pakete werden im *next-hop* Verfahren stets von Knoten zu Knoten weitergeleitet, es gibt keinen zentralen Server der als einziger Verbindungspartner für alle Sensorknoten fungiert. Bei ScatterWeb gibt es darüber hinaus (aufgrund seiner Konzeption des batteriefreien Betriebes) die Möglichkeit, Pakete entsprechend der Ausstattung der einzelnen Sensorknoten energieeffizient weiterzuleiten. Aufgrund technologiespezifischer Besonderheiten wie dem Buffer bei den Mica-Motes, entstehen Geschwindigkeits- und/oder Energievorteile. Die Anbindung an bestehende Netzwerke (Internet) erfolgt bei allen Technologien auf ähnliche Weise. Bei ScatterWeb ist dies sehr praktisch für den Benutzer gelöst. Jeder Sensorknoten reicht seine Daten (eventuell mittels anderer Sensorknoten) an einen EWS weiter, dieser stellt als Webserver die Daten über TCP in angebundene Netzwerke (Bluetooth, GPRS, Ethernet, usw.) zur Verfügung.

# 6. Verwandte Arbeiten

M. Beigl hat in seiner Arbeit *Smart-Its: An Embedded Platform for Smart Objects* [6] die Smart-Its ausführlich vorgestellt und beschrieben. J. Schiller, A. Liers und H. Ritter geben in ihrer Ausführung *ScatterWeb – Low Power Sensor Nodes and Energy Aware Routing* [2] sowohl einen Einblick in ScatterWeb als auch in das Prinzip Pakete vorzugsweise über Sensorknoten mit erneuerbarer Energiequelle weiterzuleiten. J. Hill stellt in seiner Arbeit *Mica: A Wireless Platform for Deeply Embedded Networks* [17] die Mica Technologie vor.

Eine Einführung in Sensornetzwerke und eine Bewertung anderer Technologien erfolgt in *A Survey on Sensor Networks* [23]. Vorgestellte Technologien sind hier unter anderem die µAMPS drahtlosen Sensorknoten [24] und die *wireless integrated network sensors (WINS)* Architektur [25]. In [23] wird auf verschiedene Aspekte von Sensornetzen eingegangen und auch verschiedene Protokolle auf der Sicherungsschicht, der Vermittlungsschicht, der Transportschicht und der Anwendungsschicht eingegangen sowie deren Aufgaben erläutert.

In der Arbeit *Sensor Networks Project* von A. Ali und R. Shah [26] erfolgt eine weitere Einführung in Sensornetze und die Untersuchung der Sensorknoten TmoteSky-Mote der Telos Architektur. Besonders wird hier auch auf die Fernaktualisierung des Betriebssystems und die entsprechende Weiterleitung der Updates von Sensorknoten zu Sensorknoten eingegangen

Welche Bedeutung der Zeitsynchronität in Sensornetzwerken und der physikalischen Lokalisierung von Sensorknoten in Sensornetzwerken zukommt, hat K. Römer in seiner Arbeit *Time and Location in Sensor Networks* [27] untersucht und seine Ergebnisse dargelegt.

Auf Sensornetzwerke und vornehmlich Motes und die Mica-Architektur geht auch R. Hahn und O. Neukum in *Sensornetzwerke* [28] ein. Es erfolgt eine Bewertung und Durchleuchtung von Routing-Protokollen in Sensornetzwerken und die Anforderungen an solche werden erläutert. Darüber hinaus werden Mängel und Probleme bei der Selbstkonfiguration und –organisation in Sensornetzwerken bei bestehenden Technologien betrachtet.

# 7. Zusammenfassung

In dieser Arbeit wurden drei unterschiedliche Technologien für die Realisierung von Sensornetzen vorgestellt. Jede dieser Technologien hat einen anderen Kernfokus, der als Hauptkriterium angesehen werden kann. So lässt sich bei Smart-Its die Sensormodularität und die hohe Energieeffizienz anführen. ScatterWeb verfügt bedingt durch *energy scavenging* über die Fähigkeit, einen Betrieb nicht nur durch Batterien, sondern alternativ auch durch die Verwendung von erneuerbaren Energien zu ermöglichen. Damit verbunden wurde von ScatterWeb der Begriff des *solar aware routing* (und allgemein: *energy aware routing*) geprägt, bei dem bevorzugt Sensorknoten mit erneuerbarer Energiequelle eingesetzt werden. Bei Mica-Motes spielt die Zeitsynchronisierung der Sensorknoten eine große Rolle. Darüber hinaus werden hier Algorithmen eingesetzt, die eine physikalische Lokalisierung der einzelnen Sensorknoten in den Sensornetzen zum Ziel haben. In Sensornetzen spielen viele verschiedene Faktoren und Anforderungen eine Rolle, die für die jeweiligen Anwendungsszenarien und deren Zielsetzungen durch die verschiedenen vorgestellten Technologien in unterschiedlichem Maße erfüllt werden.

Einschränkungen bezüglich des Umfanges dieser Arbeit dürfen allerdings nicht unerwähnt bleiben. Es konnten die drei

Technologien Smart-Its, ScatterWeb und Mica-Motes betrachtet und aufgrund grober Kriterien verglichen werden. Für eine weitere tiefgreifende Analyse wird auf die jeweiligen Literaturen verwiesen. Zusammenfassend ist zu bemerken, dass diese Arbeit nicht als Ratgeber dienen soll und kann, in welcher Anwendungssituation und unter welchen Umweltbedingungen, welche der drei Technologien die adäquate darstellt.

Mit dieser Arbeit sollte ein Verständis für Smart-Its, ScatterWeb und Mica-Motes erreicht werden, um diese nach Kernkonzepten, Hardwareausstattung, Zielsetzungen sowie Möglichkeiten der Anwendung zu klassifizieren. Durch diese Ausführungen sollte zudem ein Einblick in Sensornetze vermittelt werden.

Die untersuchten Technologien lassen sich nicht auf ihre scheinbaren Kernmerkmale pauschalisieren, da bei allen ein relativ hohes Maß an Anpassungsmöglichkeiten besteht. Sensornetze im Allgemeinen lassen sich nicht durch *die eine richtige* Technologie realisieren. Jedes Anwendungsszenario unterscheidet sich und bringt andere Umweltbeindungen mit sich (beispielsweise statische oder dynamische). Auch hängt es von der jeweils zu realisierenden Anwendung ab, welche Anforderungen an ein Sensornetz gestellt werden. Diese können stark variieren und sich sogar gegenseitig ausschließen (lange Lebensdauer, hohe Datenraten, schnelle Weiterleitung, hohe Energieeffizienz usw.). Aufgrund dieser Kriterien gilt es zu differenzieren und die Vor- und Nachteile entsprechender Technologien gewinnbringend einzusetzen.

## 8. Literatur


[1] Laurent Eschenauer und Virgil D. Gligor, “A key-Management Scheme for Distributed Sensor Networks,” *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41-47, 2002.

[2] Jochen Schiller et al., “ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing,” *Proceedings of the 38th Annual Hawaii International Conference on System Sciences(HICSS'05)*, pp. 1-9, 2005.

[3] Jason Hill et al., “System architecture directions for networked sensors,” *Architectural Support for Programming Languages and Operating Systems*, pp. 93-104, 2000.

[4] Heemin Park et al., “A New Light Sensing Module for Mica Motes,” *The 4th IEEE Conference on Sensors*, 2005.

[5] M. Beigl et al., “MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects,” *Computer Networks*, vol. Vol. 35, no. No. 4, pp. 401-409, 2001.

[6] Michael Beigl und Hans Gellersen, “Smart-Its: An Embedded Platform for Smart Objects,” 2003.

[7] A. Willig et al., *Altruists in the PicoRadio Sensor Network*, Västeras: Schweden, 2002.

[8] Alan Akbik und Marc Berendes, “Energy harvesting for embedded sensor/actor systems,” Technische Informatik, FU Berlin, Berlin, 2006.

[9] M. Strohbach, “The smart-its platform for embedded contextaware systems,” *Proceedings of the First International Workshop on Wearable and Implantable Body Sensor Networks*, 2004.

[10] Oliver Kasten und Marc Langheinrich, “First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network,” *Workshop on Ubiquitous Computing and Communications, PACT*, 2001.

[11] Lars Erik Holmquist et al., “Building Intelligent Environments with Smart-Its,” 2003.

[12] S. Antifakos et al., *Proactive Instructions for Furniture Assembly*: Springer, 2002.

[13] Scatterweb GmbH. "http://www.scatterweb.com - Scatterweb - wireless network solutions," 20.05.2009.

[14] S. Roundy et al., *Energy Scavenging for Wireless Networks*, Boston: Kluwer Academic Publishers, 2003.

[15] R. Shah und J. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks,” *IEEE Wireless Communications and Networking Conference*, 2002.

[16] Thiemo Voigt et al., “Solar-aware Routing in Wireless Sensor Networks,” *International Workshop on Personal Wireless Communications (PWC 2003)*, pp. 847-852, 2003.

[17] Jason L. Hill und David E. Culler, “Mica: A Wireless Platform For Deeply Embedded Networks,” 2002.

[18] A. Savvides et al., “Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors,” *7th Annual International Conference on Mobile Computing and Networking*, 2001.

[19] Young-ri Choi et al., “The mote connectivity protocol,” *Proceedings of the 12th International Conference on Computer COmmunications and Networks*, 2003.

[20] Wei Ye et al., “A Flexible and Reliable Radio Communication Stack on Motes,” 2002.

[21] G. Anastasi et al., “Performance measurements of motes sensor networks,” *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 174-181, 2004.

[22] Marco Conti et al., “Motes Sensor Networks in Dynamic Scenarios,” *International Journal of Ubiquitous Computing and Intelligence*, vol. 1, 2006.

[23] Lan F. Akyildiz et al., “A Survey on Sensor Networks,” 2002.

[24] E. Shih und et al., “Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks,” *Proc. ACM MobiCom '01*, vol. Rom, Italien, pp. 272-86, Juli 2002.

[25] G. J. Pottie und W. J. Kaiser, “Wireless Integrated Network Sensors,” *Commun. ACM*, vol. 43, no. 5, pp. 551-58, Mai 2000.

[26] Ayaz Ali und Ruta Shah, “Sensor Networks Project,” 2005.

[27] Kay Römer, “Time and Location in Sensor Networks.”

[28] R. Hahn und O. Neukum, “Sensornetzwerke,” Institut für Informatik, Ludwig Maximilians Universität, München, 2004.

# TinyOS: Ein Betriebssystem für Sensorknoten

Thomas Kothmayr
Betreuerin: Corinna Schmitt
Seminar Sensorknoten: Betrieb, Netze und Anwendungen SS2009
Lehrstuhl Netzarchitekturen und Netzdienste, Lehrstuhl Betriebssysteme und Systemarchitektur
Fakultät für Informatik
Technische Universität München
Email: kothmayr@in.tum.de

***Zusammenfassung*— Das Prinzip von Sensornetzwerken beruht auf der drahtlosen Vernetzung von einigen wenigen bis zu tausenden einzelner Sensorknoten. Da diese möglichst klein und preisgünstig sein sollen, bedeutet dies, dass nur sehr begrenzte Ressourcen zur Verfügung stehen. TinyOS wurde entwickelt um auf solcher Hardware effiziente Modularität und nebenläufige Anwendungen zu unterstützen. Die Architektur besteht aus einem eventbasiertem Kernel und statisch gelinkten Anwendungsprogrammen. Diese Einschränkungen machen TinyOS sehr ressourcensparend, jedoch existieren andere Betriebssysteme für Sensorknoten, die dynamisches Laden von Programmen unterstützen (Contiki) oder preemptives Multithreading bieten (MANTIS). Im empirischen Vergleich mit MANTIS ist TinyOS jedoch sparsamer im Speicher und Energieverbrauch.**

***Schlüsselworte*— TinyOS, Sensorknoten, Berkeley Motes, Energiemanagement, Events, Tasks, Komponenten, Multithreading, MANTIS, Contiki**

## I. Einleitung

Für Sensornetzwerke gibt es mittlerweile eine Vielzahl interessanter Anwendungsgebiete, u.a. Habitat Monitoring [1] oder Structural Health Monitoring [2]. Mit zunehmend günstigeren und kleineren Sensorknoten steigt auch die Zahl der potentiellen Anwendungen. Um eine schnellere und einfachere Entwicklung auf Sensorknoten zu ermöglichen ist eine zwischen Anwendung und Hardware gelegene Schicht, also ein Betriebssystem (engl. operating system, OS), unerlässlich. Dieses muss jedoch den speziellen Herausforderungen, die aus den begrenzten Hardware und Energieressourcen der Sensorknoten erwachsen, auf geeignete Art und Weise begegnen. Zusätzlich müssen funktionale Anforderungen an *Nebenläufigkeit*, *Netzwerkkommunikation*, *Modularität* und gegebenenfalls *Echtzeitgarantien* erfüllt werden.

Im Hinblick auf diese Gesichtspunkte wurde an der Universität Berkeley das Betriebssystem TinyOS entwickelt. TinyOS ist ein Open Source Betriebssystem und unter `http://www.tinyos.net` frei verfügbar. Dem Team um Jason Hill ist es gelungen, ein eventbasiertes OS zu entwickeln, das sehr wenig Hauptspeicher (178 Byte) [3] verbraucht und sowohl Ereignisse als auch Kontextwechsel schnell abarbeitet. Die Architektur und Designentscheidungen die dies ermöglichen sind Hauptbestandteil dieser Arbeit und werden in Kapitel 3 ausführlich dargestellt. Zunächst jedoch werden in Kapitel 2 kurz die Hardwareplattformen, für die TinyOS verfügbar ist, vorgestellt, bevor in Kapitel 4 auf die TinyOS spezifische Erweiterung nesC der Sprache C eingegangen wird. Es folgt ein Vergleich mit anderen Betriebssystemen für Sensorknoten bevor die Arbeit mit einer Zusammenfassung endet.

## II. Hardware Plattformen

Die folgenden Besonderheiten von Anforderungen an Sensorknoten, auch Motes genannt, werden von [3] beschrieben. Aus ihnen lassen sich auch Qualitätsmaßstäbe für die Entwicklung eines Sensorknotenbetriebssystems ableiten:

- *Geringe Größe und Energieverbrauch*
  Um die Vision des Smart Dust, damit sind Sensorknoten mit einem Volumen von einem Kubikmillimeter, oder weniger, gemeint, [4] realisieren zu können, oder allgemein Sensorknoten so frei wie möglich platzieren zu können, müssen diese so klein wie möglich konstruiert werden. Diese physischen Begrenzungen an, u.a. die Größe des Chips, haben unmittelbare Auswirkungen auf die zur Verfügung stehende Rechenleistung. Da auch Größe und Speicherkapazität der Energieversorgung begrenzt sind, muss das Betriebssystem besondere Vorkehrungen zum effizienten Energiemanagement treffen.
- *Günstige Herstellungskosten*
  Um Anwendungen mit tausenden von Sensorknoten rentabel zu machen, müssen diese einen möglichst günstigen Stückpreis aufweisen. Momentan liegen die Kosten für einen Mote oftmals noch bei rund 100 EUR [5], die Zielkosten jedoch bei etwa 10 Cent [6]. Mit geringerem Preis geht natürlich eine noch begrenztere Menge an Ressourcen einher.
- *Hohe Nebenläufigkeit der Operationen*
  [3], [7] führen eine hohe Anzahl an nebenläufigen Operationen als Besonderheit von Sensornetzwerken an. So ist ein Knoten oftmals damit beschäftigt, Sensordaten zu sammeln, einfache Aufbereitungen darauf auszuführen, um diese dann im Netzwerk weiter zu leiten, während er gleichzeitig Daten in einem Multihop Netzwerk empfängt oder sendet. Da wenig interner Speicher zur Verfügung steht, können diese Daten nur begrenzt zwischengespeichert werden.

- *Begrenzte physische Parallelität*
  Aufgrund der genannten Einschränkungen ist die Anzahl und Leistungsfähigkeit der Controller eines Sensorknotens begrenzt. Typischerweise werden Daten an einen einzigen Mikrocontroller weitergeleitet, anstatt, wie bei herkömmlichen Systemen, von einer Vielzahl von hierarchisch geordneten Controllern bearbeitet zu werden.
- *Vielfältige Plattformen und Anwendungen*
  Aus Energie und Kostenbetrachtungen heraus ist es erstrebenswert Motes nur mit den Hardwarekomponenten auszustatten, die für die geplante Anwendung auch wirklich benötigt werden. Außerdem besteht schon eine Vielzahl an generellen Plattformen für die TinyOS verfügbar ist. Deshalb muss das Betriebssystem einen hohen Grad an effizienter Modularität aufweisen um die jeweiligen Hardwarekomponenten möglichst einfach in der Softwareentwicklung ansprechen und zur endgültigen Anwendung kombinieren zu können.
- *Zuverlässigkeit in der Anwendung*
  Sensorknoten werden in großer Anzahl, räumlich verteilt und über einen längeren Zeitraum unbeobachtet betrieben. Dies bedeutet, dass nicht nur das Betriebssystem selbst robust sein muss, sondern auch, durch die Möglichkeit zur Entwicklung effizienter verteilter Anwendungen, Einfluss auf die Zuverlässigkeit der Anwendung hat.

### A. Berkeley Motes

An der Universität Berkeley wurden neben TinyOS auch eine Reihe von Sensorknoten entwickelt die als Berkeley Motes bekannt sind. In der nachfolgenden Tabelle sind die wichtigsten Merkmale verschiedener Motes aufgeführt. Eine

| | WeC | Dot | Mica | Mica2Dot | IRIS | TelosB |
|---|---|---|---|---|---|---|
| Jahr | 1998 | 2000 | 2002 | 2002 | 2003 | 2004 |
| CPU (MHz) | 4 | 4 | 4 | 7 | 8 | 8 |
| RAM (kB) | 0.5 | 1 | 4 | 4 | 8 | 10 |
| Verbrauch (mW) | 24 | 24 | 27 | 44 | 40 | 41 |
| Senderate (kbps) | 10 | 10 | 40 | 38.4 | 250 | 250 |
| Schnittstelle | - | - | 51pin | 19pin | 51pin | 16pin |

Tabelle I
SENSORKNOTEN MIT TINYOS UNTERSTÜTZUNG, [5], [8], [9]

nähere Beschreibung des Aufbaus und der Funktionsweise von Motes liegt außerhalb des Rahmens dieser Arbeit. Hierfür sei auf [8], [9] verwiesen.

## III. TINYOS ARCHITEKTUR

Jede TinyOS Konfiguration besteht aus dem TinyOS Scheduler und verschiedenen *Komponenten*. Diese Komponenten stehen in hierarchischer Beziehung zueinander: Sie empfangen *Events* von der darunter gelegenen Schicht und geben diese an höher liegende Komponenten weiter. Von diesen nehmen sie Befehle entgegen und stellen ihrerseits wieder Befehle an niedrigere Komponenten. Die Komponenten des Hardware Presentation Layer bilden die letzte Schicht und kommunizieren direkt mit der Hardware.

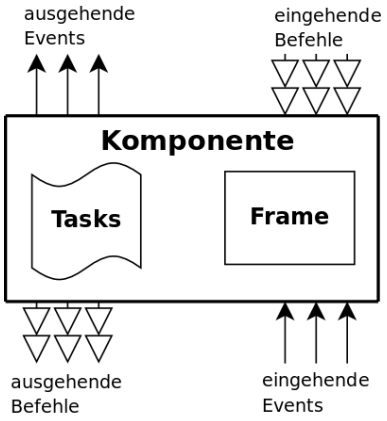



Abbildung 1. Komponente mit Schnittstellen

### A. Module und Ausführungsmodell

[3] beschreibt den Aufbau von Komponenten in TinyOS und dessen Ausführungsmodell folgendermaßen: Komponenten müssen, wie schon erwähnt, Events und *Commands* verarbeiten, weshalb sie die dafür nötigen Event- und Commandhandler besitzen müssen. Zusätzlich enthalten sie *Tasks* und einen *Frame* fester Größe der die Stati und Variablen der Tasks enthält. Um einen modularen Aufbau zu ermöglichen, werden von jeder Komponente die Events, die sie signalisiert, und die Befehle, die sie äußert, deklariert. Dies ist in *Abbildung 1* dargestellt. Hardwareinterrupts werden von Eventhandlern bearbeitet, denen die ganze Palette an Möglichkeiten einer Komponente zur Verfügung steht. Sie können Events an höhere Komponenten signalisieren, Befehle an niedrigere Module geben, Werte in den Frame schreiben oder Tasks posten, d.h. zur späteren Ausführung in die Eventwarteschlange einreihen. Befehle hingegen sind etwas eingeschränkter, denn sie dürfen keine Events auslösen, da es sonst zu Kreisen in der Command/Event Kette kommen kann. Befehle sind grundsätzlich nicht blockierend und melden das Ergebnis ihrer Ausführung als Status an das aufrufende Element zurück. Dadurch kann Polling vollständig entfallen.

Die Hauptlast an Arbeit wird von den Tasks getragen. Diese sind im Hinblick auf andere Tasks atomar, können jedoch von Events unterbrochen werden. In der Standardkonfiguration von TinyOS werden Tasks von einem einfachen FIFO-Scheduler ausgeführt, der jedoch austauschbar ist. Durch die Verwendung atomarer Tasks kann ein einzelner Stack verwendet werden, wodurch sich der begrenzte Speicher effizient nutzen lässt. Durch die Asynchronität von Events und Tasks kann Nebenläufigkeit innerhalb der Komponenten simuliert werden. Da Tasks nicht vom Scheduler unterbrochen werden, dürfen diese nicht blockieren oder aktives Warten, also die wiederholte Ausführung leerer Anweisungen bis ein gewünschter Zustand eintritt, betreiben. Ansonsten würde der Fortschritt des Gesamtsystems behindert werden.

Der Frame jedes Moduls wird statisch zugeordnet und somit stehen die Speicheranforderungen jedes Moduls bereits beim kompilieren fest. Dies bringt Vorteile in der Speicherauslastung mit sich, da eine effizientere Nutzung als bei dynamischer Zuweisung möglich ist. Für das Laufzeitverhalten sind ebenfalls positive Effekte zu verzeichnen. So ist es nicht mehr nötig Pointer aufzulösen, da die Speicherpositionen beim Kompilieren statisch festgelegt werden können.

### B. Netzwerk

Kommunikation ist zentraler Bestandteil des Konzeptes eines Sensornetzwerks. Daher beinhaltet TinyOS mehrere Protokolle und Module, um dem Anwendungsentwickler Netzwerkkommunikation auf einfache Weise zugänglich zu machen.

*a) Singlehop:* TinyOS nutzt das Prinzip von Active Messages [10] für die Singlehop Kommunikation. Dabei wird im Header eines gesendeten Pakets die Adresse eines Handlers (im Fall von TinyOS die Addresse eines Netzwerkknotens) übertragen, der bei Eintreffen der Nachricht ausgeführt wird. Die Kommunikation über Active Messages verläuft asynchron. Das Netzwerk wird dabei als Pipeline betrachtet und der Sender fährt mit seinen Berechnungen fort, nachdem er die Nachricht an das Netzwerk übergeben hat. Beim Empfänger wird beim Eintreffen der Nachricht ein Interrupt ausgelöst und der Inhalt des Pakets an einen Eventhandler übergeben. TinyOS gibt keine Zustellungsgarantien für Pakete, es handelt sich bei der Singlehop Kommunikation stets um eine best-effort Kommunikation.

*b) Multihop Routing:* Für Multihop Routing stellt TinyOS zwei grundlegende Netzwerkprimitive zur Verfügung auf denen andere Protokolle aufbauen. Dies ist zum einen *Dissemination* [11], dessen Aufgabe es ist, ein Datenpaket zuverlässig an alle Motes im Netzwerk zu verteilen. Im Gegensatz zum Flooding stellt Dissemination sicher, dass Konsistenz im Netzwerk hergestellt wird, auch wenn hohe Paketverlustraten oder temporäre Verbindungsunterbrechungen existieren. Im Gegensatz dazu wird *Collection* [12] eingesetzt um Pakete zur Wurzel eines Routingbaumes im Netzwerk zu zustellen. Falls mehrere Wurzeln existieren, werden die Pakete einer Beliebigen davon zugestellt, was einer Anycast Semantik entspricht. Dieser Baum wird durch das *Collection Tree Protocol* [13] ermittelt.

*c) Point-to-Point:* Für gewöhnlich wird Multihop Routing in TinyOS verwendet, um Energie bei der Übertragung einzusparen. Allerdings befindet sich eine TinyOS spezifische Implementierung des Dynamic MANET On-demand (DYMO) Protokolls [14] in Entwicklung. Netzwerkknoten speichern bei DYMO keine expliziten Informationen über Routing und Netzwerktopologie, sondern berechnen eine Unicast Route wenn diese benötigt wird. Dies hilft, im Vergleich mit klassischen Point-to-Point Routingverfahren, Bandbreite und Energie zu sparen, da nur sehr wenige Routinginformationen ausgetauscht werden.

### C. Energieverwaltung

Effizientes Energiemanagement ist von entscheidender Bedeutung für Sensornetzwerke. Durch die Ausnutzung des niedrigsten Schlafzustands lässt sich beispielsweise bei dem Telos Mote der Energiebedarf von 30mA auf 7-9 $\mu A$ senken. In der Praxis bedeutet dies den Unterschied zwischen wenigen Wochen und möglicherweise mehreren Jahren an ununterbrochenem Betrieb mit einem Set Batterien.

Energieeffizienz ist eines der Ziele für die TinyOS entwickelt wurde. Als grundlegenste Maßnahme versetzt TinyOS die Hardware in den Ruhezustand, sobald die Warteschlange des Schedulers keine weiteren Einträge aufweist. Jedoch haben Prozessoren verschiedene Niedrigenergielevel. Da bei einigen Anwendungsgebieten die Latenz zwischen Schlaf und Aktivzustand eine Rolle spielt, muss TinyOS das richtige Energielevel auswählen, bevor es das System in den Schlafzustand versetzt. Da die Berechnung des bestmöglichen Energiesparlevels atomar ist, sollte sie so sparsam wie möglich ausgeführt werden. In TinyOS wird dafür ein 'Dirty Bit' gesetzt, wenn eine Komponente im Hardware Presentation Layer eine Änderung an der Hardwarekonfiguration vorgenommen hat. Bevor das System das nächste Mal in einen niedrigen Energielevel versetzt wird, berechnet TinyOS dann den besten Level neu. Da einige höhere Komponenten zusätzliche Informationen haben können, die nicht allein aus dem Hardwarestatus ablesbar sind, existiert die Funktion *PowerOverride.lowestState()*. Damit lässt sich beispielsweise bei einem bevorstehendem Timerevent verhindern, dass das System in einen Zustand mit langer Aufwachzeit versetzt wird und dadurch der richtige Zeitpunkt für den Timer verpasst wird [15].

Für Peripheriegeräte werden ebenfalls Komponenten zum Energiemanagement bereitgestellt. Während in TinyOS 1.x die Applikation allein für das Verwalten des Energiebedarfs von Peripheriegeräten mittels *StdControl.start()* und *StdControl.stop()* zuständig war, wurde in TinyOS 2.x die Möglichkeit zum impliziten Energiemanagement zusätzlich geschaffen. Der *Powermanager* fungiert dabei als Standardbesitzer einer geteilten Ressource, also eines Geräts. Sobald ihm die Ressource zugewiesen wurde, kann der Powermanager seine Policy umsetzen und entweder sofort das Gerät deaktivieren oder, falls Kosten mit der Reaktivierung verbunden sind, nach einer komplexeren Regel entscheiden [16].

## IV. TinyOS Programmierung

### A. nesC

nesC ist eine Erweiterung von C deren Entwurf direkt die Konzepte von TinyOS, wie etwa Komponenten und das eventbasierte Ausführungsmodell von TinyOS unterstützt. Architektur und Design werden in [6] beschrieben. Mittlerweile ist ein großer Teil von TinyOS selbst in nesC implementiert. C selbst bietet auf der einen Hand zwar die

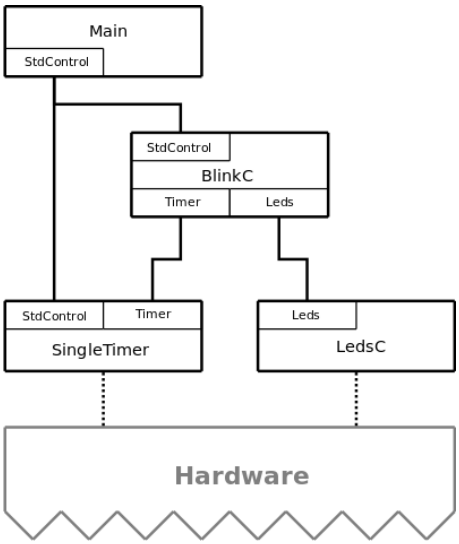



Abbildung 2. Beispielkonfiguration für eine Anwedung Blink

Möglichkeit effizienten Code für die meisten Mikrocontroller zu schreiben und erlaubt durch seine Hardwarenähe den direkten Zugriff auf die Hardware. Andererseits bietet C keine Unterstützung beim Schreiben von sicherem Code oder der Strukturierung, wie sie von TinyOS vorgesehen wird. Mit nesC hingegen wird das Komponentenmodell aktiv unterstützt und die verringerte Ausdrucksfähigkeit (z.B. statische Speicherallokation) erlaubt es, das gesamte Programm zu analysieren und zu optimieren. Dies geschieht zu Sicherheits- und Optimierungszwecken.

*Komponenten* in nesC bieten und benutzen Interfaces, die jeweils durch einen Interface Typ definiert werden. Diese Interfaces sind bidirektional, denn sie definieren zum einen Events, die an eine darüber gelegene Komponente gerichtet sind, als auch Commands, die von der darüber gelegenen Komponente aufgerufen werden können. Durch die Trennung von Interfacedefinition und Implementierung in Modulen wird die Definition von Standard Interfaces erleichtert, was Komponenten besser wiederverwertbar macht.

Neben Modulen gibt es noch eine weitere Art von Komponenten in nesC: *Konfigurationen*. Diese verknüpfen die Interfaces verschiedener Komponenten miteinander, indem sie die angebotenen Interfaces mit den benutzten Interfaces verbinden. *Abbildung 2* zeigt wie eine Konfiguration für eine einfache Anwendung aussehen kann. Jedes nesC Programm wird durch eine top-Level Konfiguration beschrieben. Diese Art der Programmorganisation ist sehr flexibel, da angebotene Interfaces 0 bis n-mal von anderen Komponenten benutzt werden können. Durch den Verzicht auf Pointer und die Verknüpfung von Modulen durch Konfigurationen wird der Programmfluss in nesC explizit, was die zuvor erwähnte Analyse des gesamten Programms ermöglicht. Ein sehr interessantes Feature von nesC ist die automatische Erkennung von Wettlaufsituationen. Code der von mindestens einem Interrupthandler aus zugänglich ist, wird dabei als asynchron betrachtet, Code der nur von Tasks aus zugänglich ist, wird, wegen der run-to-completion Ausführung in TinyOS, als synchron bezeichnet. Setzt man voraus, dass jede Aktualisierung eines geteilten Zustands (Variable) zwischen synchronem und asynchronem Code oder nur zwischen asynchronem Code in einem atomaren Block stattfindet, dann lässt sich folgende Invariante formulieren:

> *Jeder Zugriff auf einen geteilten Zustand ist entweder frei von potentiellen Wettlaufsituationen (nur synchroner Code) oder in einem atomaren Block.* [6]

Diese Invariante wird in nesC beim Kompilieren sichergestellt und verhindert so individuelle Wettlaufsituationen. Für fehlerhafte Verwendung von atomaren Blöcken können weiterhin Wettlaufsituationen entstehen. Für den Fall, dass vom Compiler als potentielle Wettlaufsituationen gemeldete Blöcke diese Gefahr nicht wirklich aufweisen, kann der Programmierer den entsprechenden Bereich mit `norace` kennzeichnen, um den Overhead eines überflüssigen atomaren Blocks zu vermeiden.

### *B. Anwendungsbeispiel: RadioSenseToLeds*

Im Nachfolgenden wird die generelle Programmierweise am Beispiel des frei verfügbaren Beispielprogramms *RadioSenseToLeds* [17] verdeutlicht. Das Programm liest einen Wert vom Standardsensor einer Plattform und überträgt ihn in einem Funkpaket. Sensorknoten die das Paket empfangen, stellen die niedrigsten 3 Bit auf ihren Leds dar. Dies setzt voraus, dass die Motes auch tatsächlich 3 Leds aufweisen, wie dies bei den Mica, Telos und IRIS Motes der Fall ist. Somit lassen sich an diesem Beispiel auch die Grundlagen von Funkkommunikation und dem Auslesen von Sensorwerten zeigen.

Das Programm besteht aus zwei Komponenten, einer Konfiguration namens *RadioSenseToLedsC.nc* und einem Modul namens *RadioSenseToLedsAppC.nc*. In der Konfiguration werden die verwendeten Module miteinander verknüpft. *Listing 1* zeigt den Quellcode der Konfiguration in *RadioSenseToLedsAppC.nc*.

```
1  #include "RadioSenseToLeds.h"
2  configuration RadioSenseToLedsAppC { }
3  implementation {
4    components MainC, RadioSenseToLedsC as App;
5    components LedsC;
6    components new DemoSensorC();
7    components ActiveMessageC;
8    components new AMSenderC(AM_RADIO_SENSE_MSG);
9    components new AMReceiverC(AM_RADIO_SENSE_MSG);
10   components new TimerMilliC();
11
12   App.Boot -> MainC.Boot;
13
14   App.Receive -> AMReceiverC;
15   App.AMSend -> AMSenderC;
16   App.RadioControl -> ActiveMessageC;
17   App.Leds -> LedsC;
18   App.MilliTimer -> TimerMilliC;
19   App.Packet -> AMSenderC;
```

```
20    App.Read -> DemoSensorC;
21 }
```

Listing 1. Die Konfiguration in RadioSenseToLedsAppC.nc

*Listing 2* deklariert die Komponente als Konfiguration. In den geschweiften Klammern können zusätzliche Interfaces benutzt (`uses` Schlüsselwort) oder angeboten (`provides` Schlüsselwort) werden. In der Toplevel Konfiguration, die in jedem Programm enthalten sein muss, und um die es sich hier handelt, kann diese Möglichkeit grundsätzlich nicht genutzt werden.

```
1 configuration RadioSenseToLedsAppC { }
```

Listing 2. Definition als Konfiguration

Die eigentliche Konfiguration befindet sich in dem mit dem Schlüsselwort `implementation` gekennzeichneten Block, von dem ein Auszug in *Listing 3* gezeigt wird.

```
1 implementation {
2   components MainC, RadioSenseToLedsC as App;
3   components LedsC;
4   components new DemoSensorC();
```

Listing 3. Components Schlüsselwort

Mittels dem `components` Schlüsselwort werden die verwendeten Komponenten angegeben und können mittels dem `as` Schlüsselwort umbenannt werden, falls mehrere Instanzen einer Komponente verwendet werden oder einfach, wie hier geschehen, um Schreibarbeit zu vermeiden. Der Rest des Codes besteht aus Verknüpfungen von Komponenten die Interfaces anbieten, mit denen die sie nutzen. Dies geschieht durch den `"->"` Operator, der rechts von der Komponente, die ein Interface mittels `uses` einfordert, steht und zu der Komponente zeigt, die das Interface anbietet (`provides`).
Die Funktionalität wird in der Datei *RadioSenseToLedsC.nc* implementiert.

```
1  #include "Timer.h"
2  #include "RadioSenseToLeds.h"
3
4  module RadioSenseToLedsC {
5    uses {
6      interface Leds;
7      interface Boot;
8      interface Receive;
9      interface AMSend;
10     interface Timer<TMilli> as MilliTimer;
11     interface Packet;
12     interface Read<uint16_t>;
13     interface SplitControl as RadioControl;
14   }
15 }
16 implementation {
```

Listing 4. Definition als Modul

Ähnlich zum Vorgehen in der Konfigurationsdatei wird in *Listing 4* mit dem `module` Schlüsselwort definiert, dass es sich beim folgenden Code um ein Modul handelt. Im Gegensatz zur Toplevel Konfiguration kann dieses Modul andere Interfaces benutzen, was hier im `uses` Block deklariert wird.

```
1  event void Boot.booted() {
2    call RadioControl.start();
3  }
4
5  event void RadioControl.startDone(error_t err)
       {
6    if (err == SUCCESS) {
7      call MilliTimer.startPeriodic(250);
8    }
9  }
```

Listing 5. Initialisierung auf Anwendungsebene

*Listing 5* zeigt wie die Funktionalität initialisiert wird. Nachdem die Plattform gestartet ist und der Eventhandler `Boot.booted()` aufgerufen wurde, ruft dieser wiederum den Befehl `RadioControl.start()` auf. Dieser wird von einer darunter liegenden Komponente bearbeitet die ein Event zurückmeldet, das in `RadioControl.startDone(error_t err)` verarbeitet wird. Im Erfolgsfall wird daraufhin der Timer mit einer Frequenz von 4 Hz gestartet. Der Timer ruft bei einem eingetretenem `fired`-Event lediglich das `read` Kommando des Sensors auf.

```
1  event void Read.readDone(error_t result,
       uint16_t data) {
2    if (locked) {
3      return;
4    }
5    else {
6      radio_sense_msg_t* rsm;
7
8      rsm = (radio_sense_msg_t*)call Packet.
           getPayload(&packet, sizeof(
           radio_sense_msg_t));
9      if (rsm == NULL) {
10 return;
11     }
12     rsm->error = result;
13     rsm->data = data;
14     if (call AMSend.send(AM_BROADCAST_ADDR, &
           packet, sizeof(radio_sense_msg_t)) ==
           SUCCESS) {
15 locked = TRUE;
16     }
17   }
18 }
```

Listing 6. Auslesen und Senden eines Messwertes

*Listing 6* zeigt das Auslesen und Senden eines Messwertes. Nachdem der Sensor einen Wert ermittelt hat, bereitet die Applikation das Senden dieses Wertes vor. Zuerst wird geprüft, ob das `locked` Flag auf `true` gesetzt ist. Dies ist der Fall, wenn das Senden des vorherigen Wertes noch nicht abgeschlossen ist. Anschließend wird der Messwert versendet. Dazu wird der Datenbereich eines Pakets extrahiert und zu einem Pointer auf den den Datentyp `radio_sense_msg_t` umgewandelt. Mittels diesem Pointer können die Felder des Pakets initialisiert werden und das Paket selbst schließlich mittels `AMSend.send` an die Broadcastadresse verschickt werden.

```
1  event void AMSend.sendDone(message_t* bufPtr,
       error_t error) {
2    if (&packet == bufPtr) {
3      locked = FALSE;
4    }
5  }
```

Listing 7. Abschluss des Sendevorgangs

Nach einem abgeschlossenen Sendevorgang, wird das `locked` Flag wieder zurück auf `false` gesetzt. *Listing 7* zeigt den dazu gehörenden Programmcode.

```
1  event message_t* Receive.receive(message_t*
       bufPtr,
2            void* payload, uint8_t len) {
3    call Leds.led1Toggle();
4    if (len != sizeof(radio_sense_msg_t)) {return
       bufPtr;}
5    else {
6      radio_sense_msg_t* rsm = (radio_sense_msg_t
         *)payload;
7      uint16_t val = rsm->data;
8      if (val & 0x0004)
9        call Leds.led2On();
10     else
11       call Leds.led2Off();
12     if (val & 0x0002)
13       call Leds.led1On();
14     else
15       call Leds.led1Off();
16     if (val & 0x0001)
17       call Leds.led0On();
18     else
19       call Leds.led0Off();
20     return bufPtr;
21   }
22 }
```

Listing 8. Empfang einer Nachricht und Schaltung der Leds

Wenn das Empfangsevent ausgelöst wird, überprüft die Anwendung ob die Länge der empfangenen Nachricht mit dem erwarteten Datentyp übereinstimmt. Falls ja wird auf das Datenfeld der Nachricht zugegriffen und ermittelt in welchen Zustand die Leds versetzt werden sollen. Der Quellcode dafür ist in *Listing 8* dargestellt.

## V. Vergleich mit anderen Betriebssystemen

TinyOS war eines der ersten Betriebssysteme das speziell für Sensorknoten entworfen wurde. Seit seiner Einführung im Jahr 2000 wurde eine Vielzahl alternativer Betriebssysteme entwickelt, von denen hier zwei exemplarisch vorgestellt werden.

### A. Contiki

Contiki wurde im Jahr 2004 am Swedish Institute of Computer Science entwickelt [18]. Die grundlegende Architektur beruht, wie bei TinyOS, auf einem eventbasiertem Kernel und einem stark modularen Aufbau von Applikationen und dem OS selbst. Jedoch unterscheidet sich Contiki in zwei Punkten deutlich von TinyOS. Zum einen unterstützt es das dynamische Laden von Modulen, zum anderen bietet es für preemptives Multitasking für einzelne Prozesse.

Um Funktionalität hinzu zu fügen oder Programmfehler zu beheben, ist es nötig neue Versionen des Programmcodes zu verteilen. Da Sensornetzwerke, entsprechend ihrem Einsatzprofil, aus potentiell hunderter einzelner Knoten bestehen, und zudem räumlich verteilt sind, erweist es sich als schwer praktikabel diese alle einzusammeln und neu zu programmieren. Daher ist die Möglichkeit Motes zur Laufzeit mit neuem Programmcode auszustatten von großer Wichtigkeit. In TinyOS wird dieses Problem mit Hilfe von Virtual Machines, wie etwa Mate [19] gelöst, da das Betriebssystem kein dynamisches Laden von Komponenten unterstützt. Dieses Vorgehen bringt Vor- und Nachteile mit sich: Einerseits nimmt die Übertragung neuen Programmcodes meist große Energieressourcen in Anspruch, denn drahtlose Kommunikation macht meist einen sehr großen Teil des Energiebedarfs eines Sensorknotens aus. Hier bieten Virtual Machines Vorteile, denn es lassen sich komplexe Anweisungen in wenigen Byte zusammenfassen und somit Energie bei der Übertragung sparen. Dies wird allerdings durch erhöhte Komplexität und Energieaufwand beim Ausführen erkauft. Besonders bei prozessorzeitintensiven Anwendungen, wie etwa kryptographischen Verfahren, ist dies ein Nachteil. Contiki bietet für diesen Einsatzzweck eine bessere Alternative, denn falls nur einzelne Module ausgetauscht werden sollen können diese zur Laufzeit neu geladen werden. Da nur der Code für die benötigten Module übertragen wird ist der Energieaufwand beim Senden gering. In der Ausführung ist kein Unterschied zur ursprünglichen Erstprogrammierung zu verzeichnen, denn der neu geladene Code wird nativ ausgeführt.

Sowohl RAM als auch ROM Speicher werden bei Contiki in zwei Teile partitioniert. In einen *Core* Bereich in dem der Kernel, der Programmlader, die am meisten genutzten Teile der Sprache mit den erforderlichen Unterstützungsbibliotheken sowie ein Kommunikationsstack mit den erforderlichen Treibern für die Kommunikationshardware, und in einen zweiten Bereich der für ladbare Programme zur Verfügung steht. Diese Aufteilung wird beim Kompilieren vorgenommen. Programme werden durch den Programmlader in den Speicher geladen, der diese entweder von direkt angeschlossenem Speicher oder vom Kommunikationsstack beziehen kann. Der Programmlader versucht zuerst ausreichend Speicher zu reservieren, falls dies fehlschlägt wird das Laden des Programms abgebrochen. Nach erfolgreichem Laden des Programms ruft der Lader die Initialisierungsfunktion des Programms auf, die einen oder mehrere Prozesse starten oder ersetzen kann.

Preemptives Multithreading wurde in Contiki als Bibliothek, die auf dem Kernel aufsetzt, implementiert. Diese ist in einen plattformunabhängigen Teil, der mit dem Kernel interagiert, und in einen von der Hardware abhängigen Teil, der das Wechseln der Stacks und die Unterbrechungsprimitive implementiert, unterteilt. Dies wird üblicherweise durch einen Timerinterrupt, der die Prozessorregister auf den Stack des

Threads schreibt und zurück zum Stack des Kernels wechselt, umgesetzt. Im Gegensatz zu normalen Prozessen, die wie bei TinyOS nur einen einzigen Stack benötigen, muss jeder Thread seinen eigenen Stack besitzen. Threads laufen so lange auf diesem Stack, bis sie unterbrochen werden oder selbst den Prozessor freigeben.

Contiki unterstützt momentan die Telos B, T-Mote Sky, ESB und MSB430 Plattformen [20].

### B. MANTIS

Im Gegensatz zu den eventbasierten Kernels von TinyOS und Contiki implementiert MANTIS [21] preemptives Multithreading auf Kernelebene. Die dadurch erreichte Nebenläufigkeit ist nützlich in Sensornetzwerken, da ein lange laufender Thread, bei gleichzeitig eintreffenden Datenpaketen, zu Pufferüberläufen und somit Paketverlusten führen kann. Außerdem wird die Programmierung für den Anwendungsentwickler durch das automatische Timeslicing bedeutend vereinfacht.

Das Design des MANTIS Kernels ähnelt klassischen UNIX-Schedulern und bietet ein Subset der Services von POSIX. Unterstützt werden prioritätsbasiertes Scheduling mit round-robin Semantik innerhalb einer Prioritätsstufe sowie Semaphore. Ähnlich wie Contiki nimmt MANTIS eine Zweiteilung des Arbeitsspeichers vor. Unterschieden wird dabei zwischen dem Speicherplatz für globale Variablen, der beim Kompilieren statisch festgelegt wird, sowie dem Rest des RAMs, der als Heap verwaltet wird. Beim Anlegen eines Threads wird Speicherplatz für dessen Stack aus dem Heap zugewiesen und wieder freigegeben, wenn der Thread endet. Die Hauptdatenstruktur des Kernel ist eine Threadtabelle die einen Eintrag pro Thread besitzt. Da diese beim Kompilieren angelegt wird ist die maximale Anzahl der Threads begrenzt, aber beim Kompilieren anpassbar. Der Standardwert beträgt 12 [21].

Der Scheduler erhält Timerinterrupts von der Hardware, um dann Kontextwechsel auszuführen. Außerdem können Kontextwechsel durch Semaphoreoperationen sowie Systemaufrufe ausgelöst werden. Diese Timerinterrupts sind die einzigen Interrupts die vom Kernel behandelt werden, alle anderen Interrupts werden direkt an die zuständigen Gerätetreiber weitergegeben. Üblicherweise gibt ein Gerätetreiber beim Eintreffen eines Interrupts einen Semaphor frei, um einen wartenden Thread zu aktivieren, der daraufhin das Event bearbeiten kann. Zusätzlich zu Treiber und Userthreads existiert ein Idlethread.

Die von MANTIS unterstützten Plattformen sind u.a. MICA2, MICAz, und TELOS B [22].

### C. Empirischer Vergleich

[23] hat einen experimentellen Vergleich von TinyOS und MANTIS durchgeführt. Dazu wurden beide Betriebssysteme auf die DSYS25 Plattform portiert und das gleiche Anwendungsprogramm auf ihnen ausgeführt. Dabei wurden folgende Ergebnisse beobachtet:

- *Speicherverbrauch*
  MANTIS benötigt etwa 30% mehr programmierbaren Speicher als TinyOS. Zusätzlich verbrauchen beide Betriebssysteme Speicherplatz für den Kernel Stack und im Fall von MANTIS für den Stack der Threads. Letztendlich entsprechen die Speicheranforderungen beider Betriebssysteme aber den besonderen Anforderungen von Sensorknoten.
- *Eventbearbeitung*
  Unter der Annahme, dass die Verarbeitung von Netzwerkaktivität eine höhere Priorität hat, als die Ermittlung von Messwerten lässt sich folgendes beobachten: Bei MANTIS ist die durchschnittliche Paketbearbeitungszeit unabhängig von der Dauer der Messwertermittlung, bei TinyOS hingegen steigt diese mit längerer Dauer der Messaktivität an. Zudem ist die Varianz der Paketbearbeitungszeit bei TinyOS signifikant größer als bei MANTIS. Somit ist Mantis besser geeignet für Szenarien bei denen ein vorhersagbares Verhalten für solche Bearbeitungszeiten nötig ist.
- *Energieverbrauch*
  Mit zunehmender Entfernung des Sensorknotens von der Wurzel steigt der Energieverbrauch, da die Menge der durch Netzwerkaktivität erzeugten Tasks ansteigt. Durch die Kontextwechsel in MANTIS steigt jedoch der Energieverbrauch bei MANTIS stärker an als bei TinyOS. Somit ist TinyOS im Allgemeinen energieeffizienter als MANTIS.

## VI. Zusammenfassung und Ausblick

TinyOS entgegnet den besonderen Anforderungen und Limitierungen von Sensorknoten, entsprechend seiner Designziele, erfolgreich. Das eventbasierte Ausführungsmodell ist geeignet, um nebenläufige Operationen, wie Netzwerkoperationen, Messwertermittlung und Aufbereitung auszuführen. Es erlaubt relativ einfache Kommunikation in Multihop Netzwerken und wird in naher Zukunft point-to-point unterstützen. Durch die Verwendung von Virtuellen Maschinen, wie etwa Mate, ist es möglich Code zur Laufzeit nach zu laden oder zu ersetzten. Im Vergleich mit preemptiven Multithreading Betriebssystemen für Sensorknoten weißt TinyOS einn geringeren Speicherbedarf und bessere Energieeffizienz auf, auch wenn die Ausführungszeit priorisierter Operationen generell unvorhersehbarer ist. Durch den modularen Aufbau, der durch nesC umgesetzt wird, und das Simulationsprogramm TOSSIM [24] ist ein relativ einfaches Entwickeln für TinyOS möglich. Somit lässt sich sagen, dass TinyOS eine sehr geeignetes Betriebssystem zur Entwicklung von Anwendungen für Sensornetzwerke ist.

## LITERATUR


[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," 2002.

[2] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Wireless sensor networks for structural health monitoring," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, pp. 427–428.

[3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *In Architectural Support for Programming Languages and Operating Systems*, 2000, pp. 93–104. [Online]. Available: http://www.cs.berkeley.edu/~culler/cs252-s02/papers/tos.pdf

[4] B. Warneke, B. Liebowitz, and K. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Classical Papers on Computational Logic*, vol. 1, pp. 372–383, 2001.

[5] Crossbow technology corporate website. [Online]. Available: http://www.xbow.com/

[6] D. Gay, M. Welsh, P. Levis, E. Brewer, R. von Behren, and D. Culler, "The nesc language: A holistic approach to networked embedded systems," in *In Proceedings of Programming Language Design and Implementation (PLDI*, 2003, pp. 1–11. [Online]. Available: http://www.cs.berkeley.edu/~pal/pubs/nesc.pdf

[7] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler, "The emergence of networking abstractions and techniques in tinyos," in *In NSDI*, 2004, pp. 1–14. [Online]. Available: http://www.cs.berkeley.edu/~polastre/papers/tinyos-nsdi04.pdf

[8] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, p. 12–24, 2002.

[9] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN*, 2005, pp. 364–369.

[10] T. V. Eicken, D. Culler, S. Goldstein, and K. Schauser, "Active messages: a mechanism for integrated communication and computation," 1992.

[11] P. Levis and G. Tolle, "Dissemination of Small Values," TinyOS Network Working Group, Tech. Rep. [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep118.html

[12] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Collection," TinyOS Network Working Group, Tech. Rep. [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep119.html

[13] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, , and A. Woo, "The Collection Tree Protocol (CTP)," TinyOS Network Working Group, Tech. Rep. [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html

[14] Dynamic MANET On-demand (DYMO) Routing. [Online]. Available: http://ianchak.com/dymo/draft-ietf-manet-dymo-12.html

[15] R. Szewczyk, P. Levis, M. Turon, L. Nachman, P. Buonadonna, and V. Handziski, "Microcontroller Power Management," TinyOS Core Working Group, Tech. Rep. [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep112.html

[16] K. Klues, V. Handziski, J.-H. Hauer, and P. Levis, "Microcontroller Power Management," TinyOS Core Working Group, Tech. Rep. [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep115.html

[17] (2005, June) RadioSenseToLeds exsample sourcecode. [Online]. Available: http://www.tinyos.net/tinyos-2.x/apps/RadioSenseToLeds/

[18] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," 2004. [Online]. Available: http://www.sics.se/~adam/dunkels04contiki.pdf

[19] P. Levis and D. Culler, "Mate: A tiny virtual machine for sensor networks," 2002. [Online]. Available: http://www.cs.berkeley.edu/~pal/pubs/mate.pdf

[20] Contiki Dokumentation. [Online]. Available: http://www.sics.se/~adam/contiki/docs/a01162.html

[21] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "Mantis os: An embedded multithreaded operating system for wireless micro sensor platforms," in *ACM/Kluwer Mobile Networks Applications (MONET), Special Issue on Wireless Sensor Networks*, 2005.

[22] MANTIS Homepage. [Online]. Available: http://mantis.cs.colorado.edu/index.php/tiki-index.php

[23] C. Duffy, U. Roedig, J. Herbert, and C. Sreenan, "An experimental comparison of event driven and multi-threaded sensor node operating systems," in *In Proceedings of the Third IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PERSENS2007), White Plains*, 2007.

[24] P. Levis and N. Lee, "Tossim: A simulator for tinyos networks," 2003.

# Einführung in das Structural Health Monitoring

Marco Antonio Volbracht
Betreuerin: Corinna Schmitt
Seminar Sensorknoten: Betrieb, Netze & Anwendungen SS 2009
Lehrstuhl Netzarchitekturen und Netzdienste, Lehrstuhl Betriebssysteme & Systemarchitektur
Fakultät für Informatik
Technische Universität München
Email: volbrach@in.tum.de

## Kurzfassung
Um die strukturelle Sicherheit von Bauwerken bzw. den dazugehörigen Bauteilen zu erhöhen, wurden bisher in regelmäßigen Intervallen Inspektionen durchgeführt. Hierbei sollen strukturelle Schäden frühzeitig erkannt und somit einem Ausfall vorgebeugt werden. Das Structural Health Monitoring (SHM) ist eine Methode zur Automatisierung dieses Prozesses. In diesem Paper wird dieses Konzept erläutert und ein Vergleich zum Condition Monitoring (CM) gezogen. Des Weiteren wird näher auf das „Golden Gate Projekt“ (GGP) eingegangen, welches dem SHM zuzuordnen ist.

## Schlüsselworte
Structural Health Monitoring, Condition Monitoring, Wireless Sensor Network, Sensor Board, Crossbow MicaZ

## 1. Einleitung
Das Problem struktureller Sicherheit von Bauwerken ist in manchen Gebieten ein weit verbreitetes Problem. Um dieses in den Griff zu bekommen wurde versucht die Identifikation von strukturellen Schäden zu automatisieren. Hierbei wurde das Konzept des SHM entwickelt. Verschiedene Sensoren, die zu einem Netzwerk zusammen geschlossen sind, liefern ständig Daten über den Zustand des Bauwerkes. Anhand dieser Daten kann dann festgestellt werden ob ein Schaden eingetreten ist oder eintreten wird.

Ein Konzept welches dagegen den Zustand von Maschinen analysiert, ist das CM.

Im folgenden Abschnitt wird nun näher auf die Ziele des SHM eingegangen. Im darauf folgenden Kapitel wird dann kurz das CM vorgestellt. Nach einem kurzen Vergleich zwischen den beiden Konzepten, wird im Abschnitt 5 das GGP und seine Funktionsweise beschrieben.

## 2. Definition und Ziele des SHM
Wie oben schon erwähnt, handelt es sich beim SHM um einen automatisierten Prozess zur Identifikation von strukturellen Schäden an Bauwerken. Schäden werden hierbei als Veränderung des Materials und/oder der geometrischen Eigenschaften dieser Systeme definiert, welche sich negativ auf die Sicherheit auswirken können. [1]

Die beiden Hauptfaktoren hierbei sind die zeitliche Veränderung, also wie schnell findet diese statt, und der Intensitätsgrad der Veränderung. Es gibt zwei SHM Lösungsansätze: zum einen die direkte Schadenserkennung (visuelle Erkennungsmethoden wie z.B. Kameras oder Röntgenstrahlen), zum anderen eine indirekte Schadenserkennung (Veränderungen der strukturellen Eigenschaften). [2]

### 2.1 Axiome des SHM
In den letzten 20 Jahren wurden im Bereich des SHM diverse Axiome (eine als Wahr angenommene Aussage) erarbeitet, die nachfolgend aufgelistet werden: [3]

1. Axiom I: Alle Materialien haben inhärente Fehler oder Defekte.
2. Axiom II: Zur Bewertung von strukturellen Schäden ist ein Vergleich zwischen zwei Systemzuständen notwendig.
3. Axiom III: Die Identifizierung und Lokalisierung von Schäden kann in einem unbewachten Lernmodus stattfinden. Die Identifizierung des Schadentyps und der Schwere des Schadens ist generell aber nur in einem bewachten Lernmodus möglich.
4. Axiom IVa: Sensoren können den Schaden nicht messen. Eine Nachbearbeitung der gewonnen Daten und eine statistische Einordnung sind notwendig um die Sensordaten in konkrete Schadensinformationen zu konvertieren.
5. Axiom IVb: Je sensibler die Messung des Schadens ist, desto sensibler regiert diese auf Veränderungen der Betriebs- und Umgebungsbedingungen. Eine intelligente Nachbearbeitung der Daten ist also zwingend notwendig.
6. Axiom V: Die Dauer- und Zeiteinheiten die im Zusammenhang mit dem Schadensbeginn und der Schadensentwicklung stehen, bestimmen die erforderlichen Eigenschaften des SHM Sensorsystems.
7. Axiom VI: Es besteht ein Trade-off zwischen der Sensibilität eines Algorithmus zur Ermittlung eines Schadens und seiner Fähigkeit zur Rauschunterdrückung.
8. Axiom VII: Die Größe der Schäden die aus der Systemdynamik ermittelt werden kann, ist umgekehrt proportional zur Erregerfrequenz.

### 2.2 Ansätze des SHM
#### *2.2.1 Ölindustrie*
In den 70er und 80er Jahren hat die Ölindustrie viele Anstrengungen unternommen um eine Vibrations-basierte SHM Technologie zur Ermittlung von Schäden an Bohrinseln zu entwickeln. Es wurden zuerst Schadensszenarien am Computer simuliert, um diese dann mit den gemessen Daten zu vergleichen. Da es viele Probleme gab, wie Schwierigkeiten bei der Messung (z.B. bedingt durch Vibrationen der Maschinen oder Veränderungen der Masse der Bohrinseln bzw. des Inhalts der Tanks) wurde in den frühen 80er Jahren die Entwicklung von SHM Technologien für Bohrinseln weitestgehend eingestellt. [5]

### *2.2.2 Raumfahrt*

Die Raumfahrt begann ebenfalls in den 70er und 80er Jahren mit der Erforschung von Vibrations-basierten SHM Technologien. Primäres Objekt der Untersuchungen ist das Space Shuttle Programm gewesen. Es wurde ein SMIS (Shuttle Modal Inspection System) entwickelt, welches zur Identifikation von Materialermüdung eingesetzt wurde. Alle seit 1987 gestarteten Raumfahrzeuge werden periodisch einem SMIS Test unterzogen. Seit den 90er Jahren wird an SHM Technologien für Verbundwerkstoffe geforscht. Grund dafür ist die Einführung von Composite-Tanks für wieder verwendbare Raumfahrzeuge. Herausforderung hierbei, ist dass das Sensorssystem keinen Ausgangspunkt für einen Funken darstellen darf. Aktuelle Forschungen im Bereich des SHM in der Raumfahrt befassen sich mit Lichtwellenleiter-basierten Sensorsystemen. [5]

### *2.2.3 Zivile Infrastrukturen*

In diesem Bereich wird seit den frühen 80er Jahren geforscht. Hauptbetrachtungsobjekte sind hierbei Brücken und Gebäude. Auch hier wird auf Vibrations-basierte SHM Technologien gesetzt. Herausforderungen sind hier die Umwelt- und Betriebsbedingungen. Des Weiteren ist oft die Größe der Struktur ein nicht zu unterschätzendes Problem. [5]

## 3. Definition und Ziele des CM

Das Konzept des CM analysiert, im Gegensatz zum SHM, den Zustand von Maschinen und nicht den von Bauwerken. Es basiert auf der kontinuierlichen Messung von physikalischen Größen (z.B. Temperatur, Schwingungen). Das CM verfolgt zwei Ziele:

1. Sicherheit: Auf Basis der analysierten Sensordaten kann ein sehr schnell reagierendes Sicherheitssystem realisiert werden. Durch eine rechtzeitige Notabschaltung kann eine eventuelle strukturelle Schädigung an der Maschine verhindert werden. Somit können kosten- und zeitintensive Reparaturen verhindert werden.
2. Maschineneffizienz: Basiert auf der Überwachung des Maschinenzustands. Es wird somit gewährleistet, dass wenn ein Bauteil bzw. mehrere Bauteile nicht mehr effizient arbeiten (d.h. die Maschine produziert z.B. zu langsam bzw. nicht in der erforderlichen Qualität), dies erkannt wird und ein Austausch dieser Bauteile erfolgt.

Die Zustandsüberwachung von Maschinen kann man in drei Teilschritte untergliedern:

1. Zustandserfassung: Hierbei handelt es sich um die Messung und Dokumentation von Maschinenparametern.
2. Zustandsvergleich: Vergleicht den Ist-Zustand mit einem vorgegebenen Referenz-Zustand. Es kann sich hierbei um einen einzuhaltenden Sollwert als auch um einen nicht zu überschreitenden Grenzwert handeln. Der Sollwert wird entweder durch vorgegebene Größen festgelegt oder bei der Maschinenabnahme ermittelt. Der Grenzwert wird meistens vom Hersteller oder Anwender der Maschine empirisch ermittelt.
3. Diagnose: Durch Zustandsvergleiche sollen möglichst früh eventuell vorhandene Fehler und deren Ursachen lokalisiert werden.

## 4. Vergleich SHM und CM

Wie in den letzten Abschnitten schon erwähnt, ist das Ziel beim SHM die frühzeitige Erkennung von strukturellen Schäden an Bauwerken zu entdecken und daraus Maßnahmen abzuleiten. Beim CM dagegen sind nicht Bauwerke das Objekt der Überwachung, sondern Maschinen. Beim SHM geht es mehr darum die Sicherheit von Bauwerken zu gewährleisten. Beim CM ist dieser Punkt ebenfalls sehr wichtig. Im Vergleich zum SHM wird aber beim CM auch stark das Ziel der Erreichung von großmöglichster Maschineneffizienz angestrebt.

## 5. Golden Gate Projekt

Bisherige Ansätze für SHM mittels Sensorbasierten Netzwerken bestanden aus vielen Beschleunigungsmessern die alle mit Platinen zur Datenerfassung verkabelt waren. Dies verursacht hohe Kosten bei der Installation und Instandhaltung des Systems. Des Weiteren kann der normale Betrieb des Bauwerkes, durch die vielen Kabel die notwendig sind, gestört werden. Zur Lösung dieses Problem wurde das Konzept eines Wireless Sensor Networks (WSN) entwickelt. Die unterschiedlichen Knoten sind hierbei nicht mehr verkabelt, sondern kommunizieren mittels eines drahtlosen Netzwerkes. Alle vorher erwähnten Nachteile eines kabelgebundenen Sensornetzwerkes werden mit der Nutzung eines WSN gelöst. Beispielsweise ist die Installation und Instandhaltung eines solchen Systems viel kostengünstiger bzw. einfacher.

Nachfolgend wird eine solches WSN für SHM, welches auf der Golden Gate Bridge (GGB) (Abbildung 1) installiert wurde, näher vorgestellt.

**Abbildung 1: Die Golden Gate Bridge bei San Francisco (http://outdoors.learnhub.com/lesson/7345-seven-wonders-of-the-modern-world-golden-gate-bridge)**

Das hier installierte WSN besteht aus 64 Sensoren, die sich entlang des 1,3 Kilometer langen Mittelstücks und dem Südturm befinden. Abbildung 2 zeigt wie die Sensoren aufgeilt sind. 56 befinden sich entlang des Mittelstücks, die restlichen 8 auf dem Südturm. Alle Sensoren messen die Umgebungs-Schwingungen mit einer Abtastrate von 1 kHz, einem Jitter (leichte Genauigkeitsschwankung) von weniger als 10µs und einer Genauigkeit von 30µG. Die erfasstem Daten werden zuverlässig über ein 46 Hop großes Netzwerk mit einer Bandbreite von 441 B/s (Bytes pro Sekunde) beim 46. Hop übertragen (Abbildung 3) [4].

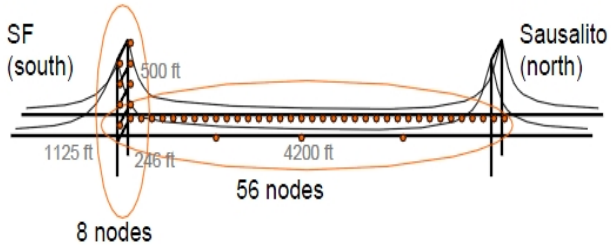



**Abbildung 2: Aufbau des WSN [4]**

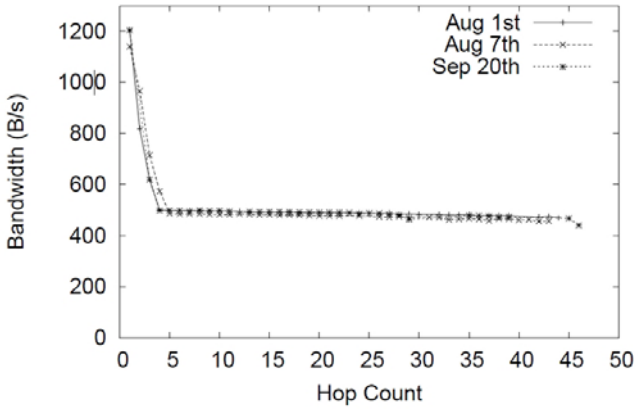



**Abbildung 3: Bandbreite und Hop Anzahl im Vergleich [4]**

## 5.1 Anforderungen

Nachfolgend werden die an die Hardware des auf der GGB installierten WSN gestellten Anforderungen aufgelistet [4]:

1. Das Daten Erfassungssystem muss in der Lage sein eine Schwingungsbreite von mindestens 500µG zu erkennen. Störquellen, wie das Grundrauschen des Systems, Installationsfehler und Variationen der Temperatur, sind zu minimieren.
2. Es wird eine Abtastrate von 1kHz angestrebt. Diese Abtastrate und die 16-Bit Genauigkeit erfordern einen geringen Jitter.
3. Zeitliche Synchronisation ist erforderlich um eine Korrelationsanalyse der Schwingungen zu ermöglichen. Da jeder Knoten seine eigene Uhr hat, wird das Flooding Time Synchronization Protokoll (FTSP) genutzt.
4. Auf Grund der großen Länge des Mittelstücks der Brücke und dass die Basisstation nur im Südturm untergebracht werden kann, wird ein großes Multi-Hop Netzwerk benötigt. Lösung für das Problem ist das MintRoute Protokoll.
5. Befehle müssen über das ganze System zuverlässig verbreitet werden können. Eine Lösung für diese Anforderung ist das Repeated Broadcast.
6. Die Daten müssen zuverlässig verteilt werden. Sie sind zu wertvoll um auf Grund von Kommunikationsfehlern verloren zu gehen.

## 5.2 Gesamtstruktur

Das WSN besteht aus vielen Knoten und einer Basisstation. Ein Knoten besteht aus einem Mote und einem Sensor Board. Der Knoten misst die Umgebungs-Schwingungen und schickt die Daten über ein drahtloses Netzwerk zur Basisstation. Diese bietet mehr Rechenleistung und Speicherkapazität als ein einzelner Knoten. Beim GGP wird als Basisstation ein handelsüblicher Laptop benutzt.

### *5.2.1 Softwarestruktur*

Wie in Abbildung 4 zu erkennen, wurden weitere Komponenten in das TinyOS (Open-Source-Betriebssystem für drahtlose Sensornetzwerke) integriert, um die in Kapitel 5.1 aufgelisteten Anforderungen zu erfüllen. Um ein System mit geringer Latenz zu ermöglichen wird bei der Verteilung von Nachrichten auf das Konzept des Repeated Broadcasting gesetzt. MintRoute wird benutzt, da es ein effizientes Multi-Hop Routing ermöglicht. Darüber befindet sich die Datenerfassungs-Schicht Straw (Scalable Thin and Rapid Amassment Without loss). Zur Zeitsynchronisation wird dagegen FTSP benutzt. BufferedLog bietet hohe Abtastraten.

Alle Komponenten werden mittels der Applikation Structural hEalth moNiToRing toolkIt (Sentri) gesteuert. Für jede Operation wird ein Befehl von der Basisstation zum jeweiligen Knoten gesendet. Sentri unterstützt 16 Operationen, wie z.B. Neustart, Löschung des Flashspeichers, Daten auslesen, usw. [4]

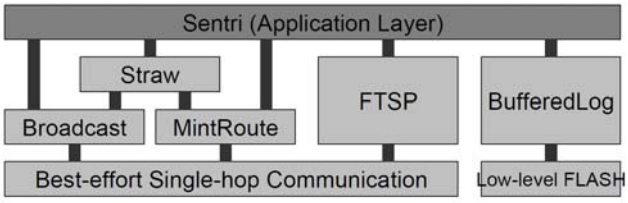



**Abbildung 4: Softwarestruktur des WSN [3]**

### *5.2.2 Hardwarestruktur*

Die Knoten im WSN erfüllen drei Hauptfunktionen: Messen, Bearbeiten von Messdaten und Kommunikation. Mit Bearbeiten der Daten ist hier die Filterung gemeint. Wie in Abbildung 5 zu sehen, besteht ein Knoten aus zwei Komponenten:

1. Sensor Board (Accelerometer Board): Besteht aus einem Thermometer und zwei Beschleunigungssensoren.
2. Mote: Kommunikations- und Kontrolleinheit.

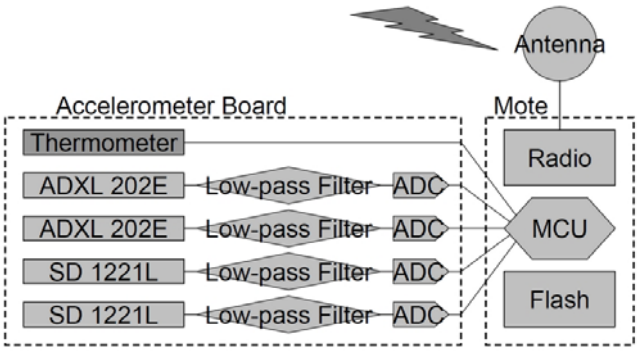



**Abbildung 5: Hardwarestruktur eines Knotens [4]**

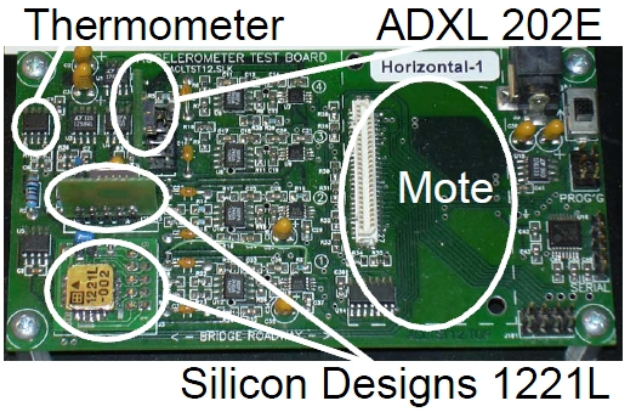



**Abbildung 6: Sensor Board mit Mote [4]**

Das Sensor Board hat zwei unabhängige Beschleunigungssensoren, die jeweils zwei Richtungen (vertikal und quer) messen. Vibrationen mit einer geringen Amplitude (z.B. Wind oder Verkehr) werden vom SiliconDesigns 1221L Accelerometer erfasst. Der kostengünstige ADXL202E Accelerometer wird dagegen zur Erfassung von stärkeren Vibrationen, wie sie z.B. bei Erdbeben entstehen, genutzt. (Abbildung 6)

Die gemessenen Daten werden gefiltert und dann von einem Analog-Digital-Umsetzer (ADC) in digitale Daten umgewandelt. [4]

Als Energiequelle stehen für jeden Knoten 4 Batterien zur Verfügung. Die Beschleunigungssensoren, der ADC und der Mote haben einen einzigen Stromanschluss. In der nachfolgenden Tabelle ist der Stromverbrauch bei jeweils unterschiedlichen Zuständen angegeben:

| *Zustand* | *Verbrauch in mW* |
|---|---|
| Nur Sensor Board | 240.3 |
| Nur Mote | 117.9 |
| Leerlauf | 358.2 |
| Eine LED an | 383.4 |
| Flashspeicher löschen | 672.3 |
| Messen | 358.2 |
| Daten senden | 388.8 |

**Tabelle 1: Stromverbrauch der verschiedenen Zustände [4]**

Wenn man nur den Mote an die Batterie anschließen würde, könnte man einen geringeren Stromverbrauch erzielen. Falls die anderen Komponenten dann nicht gebraucht werden, könnte man diese separat abschalten. [4] TinyOS unterstzütz verschiedene Engeriemodi (z.B. sleep und awake), welche es dem Knoten ermöglichen effizient mit der nur begrenz verfügbaren Energie umzugehen.

## 5.3 Zuverlässige Datenerfassung

Um eine zuverlässig Datenerfassung zu ermöglichen ist das oberste Ziel eine verlässliche und verlustfreie Kommunikation innerhalb des Netzwerkes. Die Hauptanforderungen an ein solches Protokoll sind die Kapazität und die Skalierung auf ein größeres Multi-Hop Netzwerk. Ein weiterer wichtiger Punkt ist die Minimierung der Nutzung der Netzwerkkomponenten, da die Knoten nur geringe Rechenleistung und Speicherplatz zur Verfügung stellen und beim jedem Benutzen Strom verbrauchen. [4]

**Abbildung 7: Installiertes Sensor Board, Antenne und Batterie [4]**

## 5.4 Umweltbedingte Herausforderungen

Die GGB befindet sich einem schwierigen Umfeld. Starke Winde, dichter Nebel und häufiger Regen erschweren die Konzeption und Wartung eines elektronischen Systems. Die Kombination von Nebel und starkem Wind führt zu einem schnellen Kondensieren des Meerwassers und zur Oxidation von metallischen Bauteilen. Die Hülle für das Sensor Board ist eine wasserundurchlässige Plastikbox (Abbildung 7). Auf Grund von starkem Wind mussten viele Bauteile mit Schraubzwingen und Kabeln an die Brücke befestigt werden. Aus demselben Grund wurde, durch die dadurch entstandenen Vibrationen, die Sendeleistung der Antenne merklich beeinflusst. Um diesem entgegenzuwirken, wurden auch die Antennen der Knoten mit Kabelbindern befestigt. Auf Grund der Bauweise der Brücke muss das Radio Signal nur bidirektional sein. Trotzdem wurde eine externe bidirektionale Antenne zur „Stärkung“ der Kommunikation installiert. Entlang der Brücke gibt es viele Engstellen an denen auch zahlreiche metallische Komponenten verbaut sind. Die Reichweite des Signals ist in einem solchen Umfeld natürlich schwer beeinträchtig und erlaubt den beim GGP installierten Crossbow MicaZ Motes eine mögliche Reichweite von 15 bis 30m. [4]

## 5.5 Ergebnisse

Das GGP liefert drei Beiträge für WSNs: [4]

1. Anforderungen an ein WSN wurden formuliert, so dass Daten mit einer ausreichenden Qualität für SHM ermitteln werden können. Ein genaues Datenerfassungssystem, hohe Abtastraten und synchronisiertes Sampling werden beim GGP unterstützt.
2. Das installierte System unterstützt die Installationen von weiteren Knoten und erlaubt so ein dichte Abdeckung.
3. Das System ist in einem realen Umfeld installiert und getestet worden und hilft somit zahlreiche Probleme zu lösen. Das WSN liefert zuverlässige Daten für eine Analyse, welche so vorher nicht möglich gewesen ist.

## 5.6 Vergleich zu anderen Projekten

### *5.6.1 Berkeley Fußgängerbrücke*

Ähnlich wie das GGP. Die Brücke wurde als Testobjekt für das GGP genutzt. Sie (Abbildung 8) ist 79m lang und 5m breit. Sie ist um einiges kleiner als die GGB, daher wurden nur 13 Motes angebracht (Abbildung 9). 10 davon messen die Strukturvibrationen; die restlichen 3 werden für anderweitige Tätigkeiten genutzt. Die Abtastrate beträgt 1kHz über 4 Minuten. 5 Messungen werden zu Einer zusammengefasst; damit ergibt sich eine effektive Rate von 200Hz. [2]

**Abbildung 8: Berkeley Fußgängerbrücke [2]**

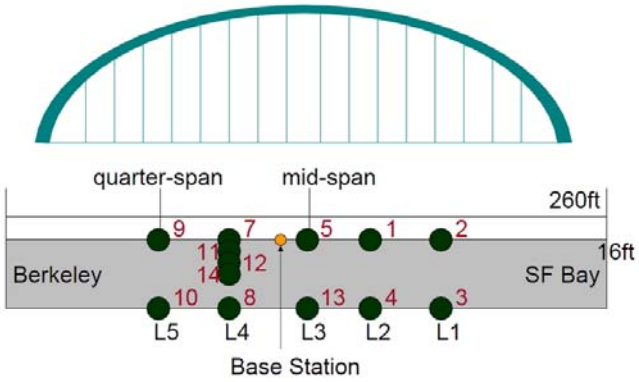



**Abbildung 9: Position der Motes (Berkeley Fußgängerbrücke) [2]**

### *5.6.2 Wind and Structural Health Monitoring System (WASHMS) für Brücken in Hong Kong*

Es handelt sich hierbei um ein umfangreiches SHM Projekt in Hong Kong welches die Brücken Tsing Ma (1.377m) (Abbildung 10), Ting Kau (1177m) und Kap Shui Mun (820m) betrifft. Das WASHMS besteht aus 4 Ebenen: Sensorsysteme, Datenerfassungssysteme, lokale zentrale Computersysteme und globale zentrale Computersystem. Es wurden 900 Sensoren installiert (ca. jeweils 350 auf der Tsing Ma und der Ting Kau Brücke und ca. 200 auf der Kap Shui Mun Brücke). Die gemessen Daten werden zum jeweiligen Datenerfassungssystem geschickt. Von dort werden diese dann an das lokale zentrale Computersystem bzw. danach an das globale zentrale Computersystem weitergeleitet. Dort findet die eigentliche Analyse der ermittelten Daten statt. Im Gegensatz zum GGP wird hier mit GPS Modulen gearbeitet. Hierbei wird eine Genauigkeit von 10mm in der Horizontalen und 20mm in der Vertikalen erreicht. Die Module befinden sich in der Nähe von anderen Sensoren (z.B. Beschleunigungsmessern) um die von ihnen ermittelten Daten besser mit denen der anderen Sensoren vergleichen zu können. Ebenfalls anders als wie beim GGP wurde hier eine kabelgebundene Lösung benutzt und nicht ein WSN. [6]

**Abbildung 10: Tsing Ma Brücke in Hong Kong (http://www.worldcountries.info/HongKong/a1-HongKong-01.htm)**

## 6. Zusammenfassung

Das SHM ist ein recht neues Konzept welches versucht automatisch die Funktionsfähigkeit von Bauwerken bzw. deren Bauteilen zu ermitteln. Dies geschieht mittels Sensoren die zu einem Netzwerk zusammengeschlossen sind. Im Gegensatz zum SHM, befasst sich das CM mit der Überwachung von Maschinen. Hierbei geht es um deren Sicherheit und die Maschineneffizienz.

Als Beispiel für den Einsatz von SHM in der Praxis wurde in diesem Paper das GGP näher vorgestellt. Hier wird ein WSN verwendet, welches geringere Kosten sowohl in der Anschaffung sowie bei der Wartung verursacht. Es handelt sich hierbei um ein wegweisendes Projekt, da es als eines der Ersten eine drahtlose und nicht kabelgebundene Lösung für SHM nutzt.

## 7. Literatur


[1] Charles R. Farrar, Keith Worden. *An introduction to structural health monitoring*. (Dezember 2006)

[2] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steve Glaser, Martin Turon. *Poster Abstract: Wireless Sensor Networks for Structural Health*. (November 2006)

[3] Keith Worden, Charles R. Farrar, Graeme Manson, Gyuhae Park. *The fundamental axioms of structural health monitoring*. (2007)

[4] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, Martin Turon. *Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks*. (April 2007)

[5] Charles R. Farrar, Scott W. Doebling, David A. Nix. *Vibration-based structural damage identification*. (2001)

[6] Kai-yuen Wong, King-leung Man, Wai-yee Chan. *Real-Time Kinematic Spans the Gap*. (Juli 2001)

ISBN 3-937201-07-6

ISSN 1868-2634 (print)
ISSN 1868-2642 (electronic)