Allgegenwärtige Kommunikation heterogener Systeme in Smart Ensembles

Dissertation
zur
Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik
der Universität Rostock

vorgelegt von Enrico Dressler
geboren am 02.11.1979 in Anklam

urn:nbn:de:gbv:28-diss2010-0146-8

Gutachter:

- Prof. Dr.-Ing. habil. Djamshid Tavangarian
  Universität Rostock, Fakultät für Informatik und Elektrotechnik
- Prof. Dr. rer. nat. Clemens H. Cap
  Universität Rostock, Fakultät für Informatik und Elektrotechnik
- Prof. Dr. phil. nat. habil. Bernd Klauer
  Helmut-Schmidt-Universität, Universität der Bundeswehr Hamburg, Fachbereich Elektrotechnik

Tag der Einreichung: 08. April 2010
Tag der Verteidigung: 06. Juli 2010

Kurzfassung

Der anhaltende Fortschritt der Mikroelektronik führt zu immer kleineren, schnelleren und energieeffizienteren Netzwerkschnittstellen, die den Einsatz in kleinen sowie spontan und drahtlos miteinander kommunizierenden Geräten (z. B. Sensoren, Mobiltelefone und Notebooks) ermöglichen. Diese Geräte erlauben eine heterogene und ubiquitäre Informationsverarbeitung mit Fokus auf die proaktive und situationsangepasste Unterstützung des Nutzers in seiner alltäglichen Arbeit. Zur Datenübertragung kommen mit Ethernet, WLAN, Bluetooth und ZigBee schon heute verschiedene drahtgebundene und drahtlose Netzwerktechnologien zum Einsatz, die sich in ihrer Reichweite, den zur Verfügung stehenden Datenraten, den Adressierungsarten, den Kommunikationsprotokollen und dem Energieverbrauch deutlich unterscheiden. Die Nutzung dieser Technologien führt gerade durch den Einsatz verschiedener IP- und nicht-IP-basierter Adressierungs- und Kommunikationsarten zu Inkompatibilitäten in heterogenen Netzwerken. Die vorliegende Arbeit untersucht und erweitert den momentanen Stand der Forschung zur Interoperabilität dieser Technologien anhand einer Referenzarchitektur eines General Purpose Access Points (GPAP) um Mechanismen, die sowohl auf der Internetschicht als auch auf der Anwendungsschicht des TCP/IP-Modells für eine allgegenwärtige Kommunikation in heterogenen Netzen führen. Die Funktionsfähigkeit des entwickelten Kommunikationskonzeptes und der realisierten GPAP-Referenzarchitektur werden in praktischen Szenarien beispielhaft evaluiert.

Abstract

The continuing progress in microelectronics leads to ever-smaller, faster and energy-efficient network interfaces that allow small as well as spontaneous and wirelessly communicating devices (e. g. sensors, mobile phones, and laptops). These devices afford a heterogeneous and ubiquitous information processing with focus on a proactive support of the user in his everyday work depending on the situation. Today, these devices utilize significantly different wired and wireless network technologies (e. g. Ethernet, WLAN, Bluetooth, and ZigBee) for data transfer varying in range, available data rates, addressing modes, communication protocols, and energy consumption. The use of these technologies results in incompatibilities between various IP and non IP-based addressing and communication methods in heterogeneous networks. This thesis evaluates and extends the current state of research in the field of interoperability of different network technologies on the basis of a reference architecture called General Purpose Access Point (GPAP). The GPAP provides mechanism for ubiquitous communication in heterogeneous networks on the Internet layer and on the application layer of the TCP/IP model. The operability of the developed communication concept and the realized GPAP reference architecture will be evaluated exemplary in practical scenarios.

Danksagung

Diese Dissertation entstand in meiner Zeit als Promotionsstudent am Lehrstuhl für Rechnerarchitektur der Universität Rostock. Der praktische Rahmen und die Anregung für die Arbeit entstanden im Umfeld der vielfältigen Aktivitäten des Lehrstuhls für Rechnerarchitektur im Bereich drahtloser Netzwerke und insbesondere in Kooperation mit dem Graduiertenkolleg *Multimodal Smart Appliance Ensembles for Mobile Applications* (MuSAMA). Ohne diese Basis sowie individuelle Beiträge von verschiedener Seite wäre diese Arbeit nicht möglich gewesen.

An erster Stelle und ganz besonders möchte ich meinem langjährigen Chef und Mentor Prof. Djamshid Tavangarian für die produktive und angenehme Zusammenarbeit Dank sagen, in der er mir jederzeit bei Fragen und Problemen zur Seite stand. Ich danke weiterhin Prof. Clemens Cap und Prof. Bernd Klauer für ihre Bereitschaft zur Begutachtung dieser Arbeit und die daraus resultierenden Hinweise. Für die fachlichen und inhaltlichen Anmerkungen sowie das sorgfältige Korrekturlesen danke ich Dr. Ulrike Lucke und Dr. Daniel Versick herzlich.

Den zahlreichen Beteiligten am MuSAMA-Projekt gilt mein aufrichtiger Dank für die produktive und angenehme Zusammenarbeit. Weiterhin danke ich allen Mitarbeiterinnen und Mitarbeitern des Lehrstuhls für Rechnerarchitektur. Ein ganz herzlicher Dank gilt den Diplomanden Jan Miller, Friedrich Meincke, Philipp Lehsten und Tom Reichelt, für ihr Engagement und ihre Mithilfe bei der Entwicklung des GPAP-Systems und der Realisierung verschiedener Einsatzszenarien.

Keine Arbeit gelingt ohne Unterstützung von Freunden und Familie. Viele Freunde sind bereits genannt worden. Besonderer Dank geht an dieser Stelle an meine Freundin Susan, die mir immer Rückhalt und Ansporn gab, auch wenn die Motivation nachließ. Meinen Eltern bin ich sehr dankbar für das solide Fundament an Werten und Überzeugungen, das sie in mir legten. Ohne diese Unterstützung und den Rückhalt in den Jahren meiner Ausbildung wäre diese Arbeit nicht möglich gewesen.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Die seit über 40 Jahren andauernde Entwicklung des Internets hat bis zu seiner heutigen Form zu tiefgreifenden Veränderungen für das gesellschaftliche und wirtschaftliche Leben in den Bereichen Forschung, Entwicklung, Ausbildung, Informationsbeschaffung und globalem Wissensaustausch geführt [1]. Mit dem Internet steht uns nicht nur eine Art digitale Bibliothek zur Verfügung, die das Weltwissen bis zu unserer Zeit enthält und es für jedermann ohne örtliche oder zeitliche Grenzen zur Verfügung stellt, es werden uns auch Möglichkeiten der globalen Kommunikation und Kollaboration bereitgestellt, die noch vor kurzem undenkbar waren. Mit der rasanten Ausbreitung des Internets — angefangen von stationären Computern, über mobile Geräte (wie Laptops, Personal Digital Assistants (PDAs) und Mobiltelefone), bis hin zu Sensoren in unserer Umgebung — stellt das bereits begonnene Internet der Dinge, also quasi die Verlängerung des Internets bis in die letzten Alltagsgegenstände hinein, einen wesentlichen Katalysator unserer zukünftigen technologischen Entwicklung dar [2][3].

Der Grundstein für das Internet, so wie wir es heute kennen, wurde schon in den 1960er Jahren von einer kleinen Forschergruppe unter der Leitung des *Massachusetts Institute of Technology* (MIT) und des US-Verteidigungsministeriums als sogenanntes *Advanced Research Projects Agency Network* (ARPANET) entwickelt [4]. Das ARPANET verband die beteiligten Hochschulen zunächst über Telefonleitungen miteinander, um die damals noch recht knappen Rechenkapazitäten durch Datenaustausch besser ausnutzen zu können. Ebenfalls ermöglichte es erstmals den Informationsaustausch zwischen den Wissenschaftlern per E-Mail, was seinerzeit die wichtigste Applikation zur Nutzung des Netzes darstellte. Vor allem sorgte es aber durch die Nutzung spezieller Netzwerkprotokolle, die durch die *Requests for Comments* (RFC) [5] der *Internet Engineering Task Force* (IETF) als eine Reihe von technischen und organisatorischen Dokumenten beschrieben und durch allgemeine Akzeptanz und Gebrauch zu Standards wurden, für eine einheitliche Möglichkeit

über weite Strecken zu kommunizieren. Der bedeutendste Standard unter ihnen ist das *Internet Protocol* (IP), das im Jahr 1981 als RFC 791 [6] definiert wurde und bis heute weltweit als Grundlage der Kommunikation im Internet eingesetzt wird.

Erst das im Jahre 1989 von Tim Berners-Lee im europäischen Forschungszentrum CERN entwickelte *World Wide Web* (WWW) [7][8] und die ersten von anderen Forschungsgruppen realisierten grafikfähigen Webbrowser Anfang der 1990er Jahre haben wesentlich zum Wachstum und zur Popularität des Internets beigetragen, da nun auch Laien auf das Internet zugreifen konnten. Mit wachsender Zahl der Nutzer kamen neben Webseiten zur Selbstdarstellung von Personen und Firmen vor allem kommerzielle Angebote im Internet hinzu, die durch den Online-Handel (E-Commerce) im Zuge der New Economy zum Ende des letzten Jahrtausends ergänzt wurden [1].

Im Jahre 1992 ging das *Global System for Mobile Communications* (GSM) als weltweiter Mobilfunkstandard zur Sprachübertragung in Betrieb, der mit der Erweiterung *General Packet Radio Service* (GPRS) erstmals eine paketorientierte Datenübertragung mit Einsatz des IP-Protokolls vom Internet bis hin zum mobilen Gerät unterstützte und mit der gleichzeitigen technischen Entwicklung mobiler, leistungsfähiger Geräte dem steigenden Bedürfnis der Anwender nach mobilem Internet-Zugang in Ansätzen gerecht werden konnte [9]. Mit der Weiterentwicklung der Mobilkommunikation über die GPRS-Erweiterung *Enhanced Data Rates for GSM Evolution* (EDGE) bis hin zum neuen, parallel betriebenen *Universal Mobile Telecommunications System* (UMTS) [10] wurden ab dem Jahr 2003 deutlich höhere Datenraten ermöglicht. Parallel dazu entwickelte sich ab 1995 mit dem *Wireless LAN* (WLAN) [11] eine weitere stabile und robuste Funktechnologie, die das weltweit frei nutzbare 2,4 GHz *Industrial, Scientific, and Medical* (ISM)-Frequenzband verwendet, deutlich höhere Datenraten bereitstellt und problemlos in bestehende Netzwerkumgebungen integriert werden kann. Damit gestaltete sich auch die mobile Erweiterung des Internets als ein hybrides Miteinander aus Mobilfunknetzen und WLANs, das für den Nutzer je nach gerade benötigter Reichweite, Latenz und Übertragungsrate für eine mobile, ortsunabhängige Datenkommunikation genutzt werden kann [12]. Mit der einfachen und kostengünstigen Installation von WLANs wurde es den Nutzern des Internets nicht nur ermöglicht, die Informationsbestände im Internet zu nutzen und selbst zu entwerfen, sondern auch ihren technischen Zugang zum Internet aktiv mitzugestalten.

Als weltweites Netzwerk mit immer weiter steigenden Bandbreiten und gleichzeitig sinkenden Preisen zur Teilnahme daran, entwickelte sich das Internet in einem Zeitraum von etwa 40 Jahren zum Standard für die Verbreitung von Informationen jeglicher Art. Die derzeitige Verfügbarkeit von Pauschaltarifen für Telekommunikati-

onsdienstleistungen (Flatrates) ermöglicht neben der Nutzung von E-Mails, Instant Messaging, Suchmaschinen und WWW aber auch die Verbreitung zunehmend größerer Datenmengen für *Voice over IP* (VoIP)-Telefonie, Radio, Fernsehen, Video-on-Demand und File Sharing in Peer-to-Peer-Netzen. Mit der Entwicklung des Internets geht aber auch eine Veränderung seiner Nutzer einher, die ohne örtlich oder zeitlich bedingte Grenzen vom passiven Medienkonsumenten zum aktiven Web 2.0-Autor werden, der sich zu vielerlei Themen in Online-Communitys, Wikis, Blogs, Online-Spielen und virtuellen Welten (z. B. Second Life) mit Gleichgesinnten vernetzt und die klassische, bisher techniklastige Netzkultur ergänzt.

Bei näherem Blick auf den Anteil der Nutzer einiger Anwendungen, wie er in Abbildung 1.1 am Beispiel von Europa dargestellt ist, zeigt sich, dass neben der Informationssuche gerade in den Bereichen E-Mail, Soziale Netzwerke, Instant Messaging und dem Gedankenaustausch in Foren eine herausragende Bedeutung des Internets liegt. Die klassischen Informationsmedien wie Fernsehen, Radio und Nachrichten verlieren dabei mit der Entwicklung des Internets und dessen gerade jungen Nutzern zunehmend an Attraktivität [13].

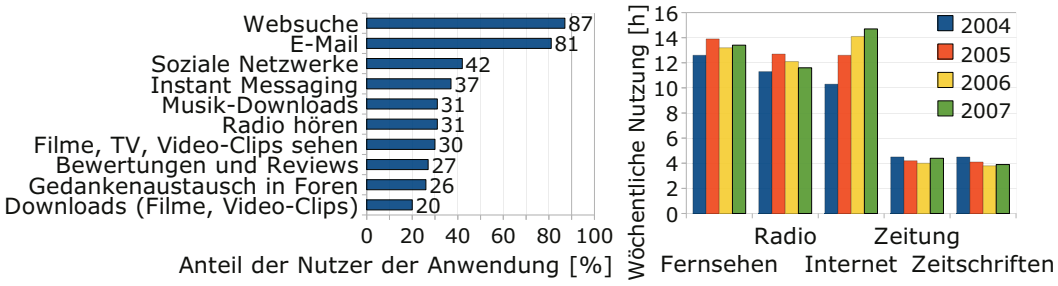

Abbildung 1.1 Art und Dauer der Internet-Nutzung in Europa [13] (Stand: 30. Juni 2009)

Die relativ einfache und zunehmend mobile Nutzbarkeit des Internets und die mit diesem Netz einhergehenden Möglichkeiten führen dazu, dass heute weltweit knapp 1,7 Milliarden Menschen das Internet nutzen [14] (Stand: 30. Juni 2009). Die Abbildung 1.2 zeigt die weltweite differierend starke Nutzung des Internets und dass gerade Europa und Nordamerika die Regionen darstellen, in denen die Menschen das Internet nicht nur zahlenmäßig, sondern auch gemessen an der Bevölkerung, am meisten nutzen. In anderen Bereichen der Erde, wie z. B. Afrika und im Nahen Osten, wird das Internet bisher weniger eingesetzt, wobei aber gerade hier die Nutzung in den nächsten Jahren noch deutlich steigen wird [15]. Andere Studien zeigen, dass die mobile Nutzung des Internets in den letzten Jahren stark angestiegen ist. Die traditionellen Informationsmedien verlieren durch das Internet zunehmend an Attraktivität [16][17].

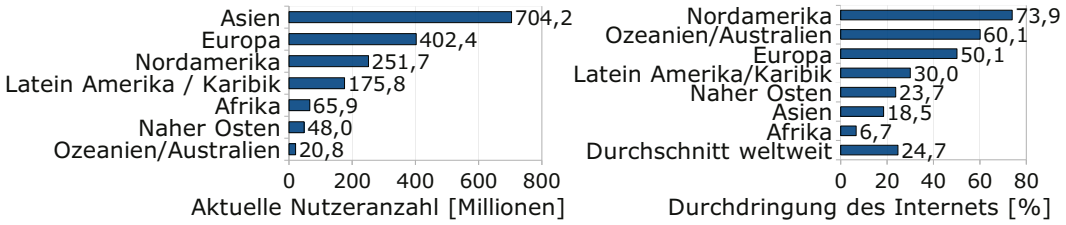

Abbildung 1.2 Weltweite Nutzungs- und Durchdringungsrate des Internets [14] (Stand: 30. Juni 2009)

Auch wenn mittlerweile so gut wie alle *Personal Computer* (PC) der Welt drahtgebunden oder drahtlos an das Internet angeschlossen sind und so selbst ein Teil dieses Netzes werden, ist die Entwicklung des Internets an sich noch lange nicht beendet. Der nächste drastische Wandel betrifft nicht die Anbindung der Geräte an das Internet, die wir täglich bewusst nutzen um auf Informationen zuzugreifen oder mit anderen Menschen über das Internet zu kommunizieren, sondern es betrifft genau die Gegenstände, die sich im Lebensalltag in unserer unmittelbaren Umgebung befinden und nur passiv wahrgenommen werden. Genauer gesagt handelt es sich um kleinste, energieeffiziente und preiswerte Computer, die sich zunehmend in unseren alltäglichen Gegenständen verstecken. Ausgestattet mit integrierter drahtloser Kommunikationsfähigkeit und Sensoren, die als „Sinnesorgane“ smarter Dinge vielfältige Umgebungsinformationen wahrnehmen, haben sie das technische Potential, spontan ein drahtloses Kommunikationsnetz zu bilden, miteinander Umgebungsinformationen auszutauschen und durch Aktoren auf spezielle Ereignisse in der Umgebung zu reagieren. Dazu benötigen sie nicht einmal eine existierende Infrastruktur oder feste Basisstationen.

Damit diese smarten Alltagsdinge den Nutzer in seiner Umgebung auch unterstützen können, reicht es jedoch nicht aus, dass hochgradig miniaturisierte Funksensoren ihre wahrgenommenen Umgebungsbedingungen einfach nur austauschen. Für den Nutzer hat ihre intelligente Zusammenarbeit zur Unterstützung seiner aktuellen Aktivität oberste Priorität, ohne ihn davon zu sehr abzulenken. Smartes, intelligentes Verhalten bedeutet in diesem Zusammenhang ein kontextbezogenes, situationsangepasstes Verhalten der Geräte. Für die Alltagsgegenstände bedeutet es, dass sie wissen, wo sie sich gerade befinden, welche Dinge oder Personen in der Nähe sind und was in der Vergangenheit mit ihnen geschah. Es bedeutet aber auch, dass sie dieses Wissen mit anderen Dingen in ihrer Umgebung teilen und auf vorgegebene Verfahren zurückgreifen, um durch kooperatives Verhalten einen Plan zur Unterstützung der aktuellen Aktivität des Nutzers zu generieren und auszuführen.

Mit der Kombination des heutigen Internets und smarten Gegenständen, die sich zunehmend in unserer alltäglichen Umgebung befinden und uns in unseren Aktivitäten assistieren, werden die Voraussetzungen für eine „totale Informatisierung“ der Welt geschaffen, die es erlaubt, Informationsverarbeitung und Kommunikationsfähigkeit in jegliche Gegenstände zu integrieren, auch in die, die zumindest auf den ersten Blick keine elektronischen Geräte darstellen. Ihr ganzes Potenzial können smarte Dinge jedoch erst durch Vernetzung und Einbindung in die umfassende Struktur und die Dienste des Internets ausspielen. Damit steht dem Internet also quasi eine Verlängerung bis in die letzten Alltagsgegenstände hinein bevor und dürfte zu diesem *Internet der Dinge* [3] noch einen drastischen Wandel erleben, der uns vor große Herausforderungen stellt und unser zukünftiges Leben mitbestimmt. Zu diesen Herausforderungen, den sogenannten *Grand Challenges der technischen Informatik* [18], von denen eine besonders hohe wirtschaftliche Bedeutung erwartet wird, zählt z. B. das Anwendungsgebiet der *zukünftigen Kommunikationsnetze*, die hinsichtlich ihrer Technologien, logischen Strukturen, Protokolle und Effizienz weiter entwickelt werden müssen. Ein Anwendungsgebiet mit gleich hoher Bedeutung ist der Bereich der *omnipräsenten Informationsverarbeitung*, der sowohl die Interoperabilität und Kooperation von Sensorsystemen als auch deren Integration in das allgemein verfügbare Internet fördert und gleichzeitig den verantwortungsbewussten Umgang mit Energie berücksichtigt.

1.2 Drahtgebundene und drahtlose Netzwerktechnologien

Technologisch gesehen umgibt uns heute eine Vielzahl unterschiedlicher drahtgebundener und drahtloser Kommunikationstechnologien, die sich je nach Art der Datenübertragung (elektrisch, optisch oder elektromagnetisch), in ihrer Reichweite und den zur Verfügung stehenden Datenraten teilweise deutlich unterscheiden. Um diese existierende Vielfalt zu verdeutlichen, stellt Abbildung 1.3 die am weitesten verbreiteten Kommunikationstechnologien schematisch hinsichtlich ihrer Reichweite und der von ihnen unterstützten Datenrate dar, bevor sich die nächsten Abschnitte der kurzen Vorstellung der für diese Arbeit relevanten Technologien widmet.

1.2.1 Local Area Networks

Neben dem ARPANET, das in den 1960er Jahren damit begann, die ersten Hochschulen miteinander zu verbinden und den Informationsaustausch zwischen den Wissenschaftlern zu fördern, entwickelten sich auch lokale Firmennetzwerke, die als sogenannte *Local Area Networks* (LAN) die technischen Einrichtungen einer Abteilung

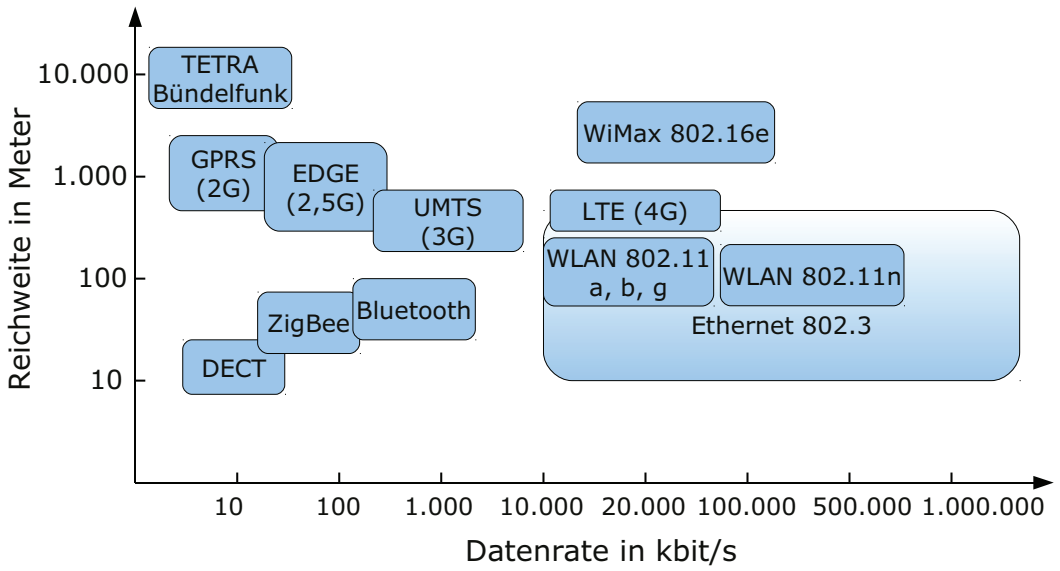

Abbildung 1.3 Beispiele heutiger Kommunikationstechnologien

bzw. eines Firmengebäudes zu einem Netzwerk verbanden und nicht nur den örtlich begrenzten Datenaustausch zwischen den Mitarbeitern sondern auch die Nutzung von Spezialhardware wie z. B. Drucker oder Plotter vereinfachte.

Die wichtigste und heute am weitesten verbreitete kabelgebundene Netzwerktechnologie für LANs stellt das ursprünglich am *Xerox Palo Alto Research Center* (PARC) in den 1970er Jahren entwickelte Ethernet dar. Es wurde am Anfang als ein firmenspezifisches und nicht standardisiertes Produkt mit einer Übertragungsrate von 3 Mbit/s entwickelt und 1976 erstmals veröffentlicht [19]. Im Jahre 1980 von der IEEE in der Arbeitsgruppe 802 standardisiert, wurden drei Techniken weiterverfolgt: *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) (802.3), *Token Bus* (802.4) und *Token Ring* (802.5), wobei die letzten beiden Techniken unter einer wahren Flut von Ethernet-Produkten als Nischenprodukte in Spezialgebiete verdrängt wurden und Ethernet nach dem Standard IEEE 802.3 [20] ab den 1990er Jahren weltweit zur ersten Wahl für lokale Netzwerke wurde. Die Standardisierung umfasst neben Festlegungen für Kabeltypen und Steckern sowohl die Spezifikation der Bitübertragungsschicht (Physical Layer) als auch der Sicherungsschicht (Data Link Layer) des ISO-OSI-Referenzmodells [21]. Dabei werden maximal 1500 Bytes an Nutzdaten in Frames verpackt und drahtgebunden im Zeitmultiplex übertragen. Hiermit kann Ethernet die Basis für Netzwerkprotokolle, wie z. B. der heute weit verbreiteten TCP/IP-Protokoll-Familie [22], bilden. Mit der technischen Weiterentwicklung der Netzwerkschnittstellen und der Verkabelung konnten die durch Ethernet bereitgestellten Datenraten stufenweise von 10 Mbit/s über 100 Mbit/s und 1 Gbit/s,

bis hin zu 10 Gbit/s (802.3ak/an/ae) erhöht werden. Mittlerweile trägt der Einsatz von Glasfaserkabeln, als Ersatz bisheriger Kupfer-basierter Kabel, auch zur Erhöhung der Reichweite von maximal 100 m auf mehrere Kilometer bei.

1.2.2 Wireless Local Area Networks

Mit dem zunehmenden Trend der Benutzer- und Endgerätemobilität sowie der damit einhergehenden flexibleren Arbeitsgestaltung entwickelte sich ab 1995 neben dem Ethernet-Standard 802.3 für drahtgebundene lokale Netzwerke mit dem allgemeinen Standard 802.11 auch ein drahtloses Pendant mit dem Namen *Wireless Local Area Network* (WLAN) [11]. Dieser Standard ist zum Ethernet-Standard kompatibel und in bestehende drahtgebundene Netzwerkumgebungen integrierbar. Zum Zugriff auf das Übertragungsmedium wird bei WLAN das *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA)-Verfahren eingesetzt. Aufgrund der Datenübertragung per Funk sind WLANs jedoch nicht mehr an Gebäude- oder Werksgrenzen gebunden, wodurch Daten auch von anderen in der Nähe befindlichen Endgeräten empfangen werden können und generell durch Sicherungsstechniken wie *Wi-Fi Protected Access 2* (WPA2), *Virtual Private Network* (VPN) oder *Internet Protocol Security* (IPSec) vor unberechtigtem Zugriff geschützt werden sollten [23].

Mit der andauernden Entwicklung von WLAN wurde der 802.11 Standard vor allem durch die Standards 802.11b, 802.11a, 802.11g und 802.11n deutlich erweitert, die mit dem *Orthogonal Frequency Division Multiplex* (OFDM)-Verfahren zwar das gleiche Modulationsverfahren einsetzen und damit mehrere orthogonale Trägersignale zur digitalen Datenübertragung nutzen können, sich aber im verwendeten Frequenzband, der Anzahl ihrer Übertragungskanäle und den von ihnen zur Verfügung gestellten Datenraten deutlich unterscheiden. So verwendet der 802.11b und der 802.11g Standard das weltweit frei nutzbare 2,4 GHz *Industrial, Scientific, and Medical* (ISM)-Frequenzband mit 11 Kanälen (davon 3 überlappungsfrei), um Daten mit 11 Mbit/s bzw. 54 Mbit/s zu übertragen. Der 802.11a Standard nutzt dagegen das 5 GHz-Frequenzband mit 19 Kanälen um eine Bruttodatenrate von 54 Mbit/s zu ermöglichen. Gegenüber diesen Standards kann der neue 802.11n Standard sowohl im 2,4 GHz- als auch im 5 GHz-Basisband eingesetzt werden. Um eine Datenrate von bis zu 300 Mbit/s zu erreichen, setzt der neue Standard zum einen auf die Verdopplung der Funkkanal-Bandbreite von 20 MHz auf 40 MHz und zum anderen auf den gleichzeitigen Einsatz mehrerer Sende- und Empfangsantennen in unterschiedlichen Abstrahlrichtungen nach der *Multiple Input Multiple Output* (MIMO)-Technik [24]. Diese für mobile Kommunikation eher hohen Datenraten stellen jedoch nur Bruttowerte dar; ihre entsprechenden Nettowerte liegen etwa bei der Hälfte der angegebenen Datenrate und teilen sich weiterhin in die Datenrate für den Up- und Downlink auf.

Mit einer maximalen Sendeleistung von 100 mW im 2,4 GHz-Band und 500 mW im 5,4 GHz-Band stellen WLANs zwar die mobile Netzwerktechnologie mit dem höchsten Energieverbrauch der in dieser Arbeit näher betrachteten Technologien dar, sie bieten aber gleichzeitig die höchsten Datenraten.

Der 802.11 Standard stellt aber auch sehr flexible Möglichkeiten zum Aufbau eines drahtlosen lokalen Netzes bereit, die durch unterschiedliche Kommunikations-Modi unterstützt werden:

Im **Infrastruktur-Modus** übernimmt ein *Access Point* (AP) die Koordination aller anderen Netzknoten. Dazu sendet er in kurzen Intervallen kleine Beacons, die einen *Service Set Identifier* (SSID) als Kennung des Funknetzwerks, eine Liste der unterstützten Übertragungsarten und die Art der Verschlüsselung enthalten. Damit ermöglicht der AP es den anderen Netzknoten, die Empfangsqualität kontinuierlich zu überwachen, auch wenn keine Nutzdaten gesendet oder empfangen werden. Aufgrund der Koordination durch den AP entsteht eine Art Sterntopologie mit dem AP in der Mitte, in der jegliche Daten die zwischen den anderen Netzknoten ausgetauscht werden sollen, über den AP als Zwischenknoten übertragen werden müssen. Der AP verfügt neben einer WLAN-Schnittstelle im Allgemeinen auch über eine Ethernet-Schnittstelle, die den Anschluss an ein kabelgebundenes LAN (bzw. per WLAN-Router an das Internet) ermöglicht, wodurch dem Aufbau räumlich großer WLANs mit mehreren APs und unterbrechungsfreiem Wechsel der mobilen Netzknoten zwischen diesen APs nichts mehr im Wege steht.

Der **Bridge-Modus** ermöglicht es, zwei räumlich getrennte WLANs, die sich im Infrastruktur-Modus befinden, durch die Nutzung jeweils eines APs über eine Richtfunkverbindung zu einem WLAN zu verbinden.

Im **Ad-hoc-** bzw. **Mesh-Modus** ist keine zentrale Instanz zur Koordination des WLANs vorgesehen. In diesem Modus sind alle Netzknoten gleichberechtigt an der Koordination des Netzes beteiligt. Sie benutzen alle dieselbe SSID und optional dieselben Verschlüsselungseinstellungen, womit sich Ad-hoc-Netze schnell und ohne großen Aufwand aufbauen lassen. Da die Weiterleitung der Daten zwischen den Netzknoten im 802.11 Standard nicht vorgesehen ist, sind spezielle Ad-hoc-Routingprotokolle nötig, um einen Überblick über die Netzwerktopologie und darauf aufbauender Entscheidungen für die Weiterleitung der Nutzdaten geben zu können. Als wichtigste Routingprotokolle in diesem Bereich sei an dieser

Stelle auf *Optimized Link State Routing* (OLSR) [25] und *Ad-hoc On-demand Distance Vector* (AODV) [26] verwiesen, mit denen sich WLAN-Ad-hoc-Netze, wie z. B. im Freifunk-Projekt [27] mit einer Größe von mehreren Quadratkilometern (allein in Berlin) im täglichen Einsatz befinden, die von immer mehr Bürgern in Eigenregie aufgebaut und gewartet werden.

1.2.3 Mobilfunknetze

Nachdem GSM im Jahre 1992 um GPRS erweitert wurde und so erstmals eine paketorientierte Datenübertragung mit einer theoretischen Datenrate von 171,2 kbit/s in Mobilfunknetzen unterstützte, begann auch in diesen Netzen ein steigendes Bedürfnis mobil auf das Internet zugreifen zu können. Dabei wurden Daten aus einer Mischung von Frequenz- und Zeitmultiplexing über mehrere Kanäle in den Bereichen 900 MHz (GSM 900) und 1800 MHz (DCS 1800) übertragen. Durch die Erweiterung von GPRS durch EDGE, konnten diese sogenannten 2,5G-Netze zwar die Datenrate auf bis zu 236,8 kbit/s steigern, einen wirklichen Durchbruch bei der mobilen Internet-Nutzung brachte jedoch erst das seit 2003 betriebene UMTS als Mobilfunkstandard der 3. Generation (3G). UMTS wurde ursprünglich vom *European Telecommunications Standards Institute* (ETSI) im Standard *International Mobile Telecommunications-2000* (IMT-2000) festgelegt und wird heute vom *3rd Generation Partnership Project* (3GPP) [28] weitergepflegt. Unter Verwendung des weniger störanfälligen *Wideband CDMA* (WCDMA)-Verfahrens nutzt UMTS 5 MHz breite Frequenzbänder im Bereich von 1920–1980 MHz für den Uplink und 2110–2170 MHz für den Downlink. Als datenoptimierte Netze standardisiert, ermöglichen 3G-Netze den Einsatz von IP vom Internet bis hin zum mobilen Gerät. Sie erreichen aber auch deutlich höhere Übertragungsraten von bis zu 384 kbit/s und beseitigten dadurch Kapazitätsprobleme bei der Sprach- und Multimediaübertragung. In Europa findet derzeit ein starker Ausbau der 3G-Netze statt, wodurch die Zahl der UMTS-Anschlüsse allein im Jahr 2009 um 36 Prozent auf rund 172 Millionen zugenommen hat, während die Zahl der herkömmlichen Anschlüsse auf Basis der GSM-Technologie mit 5 Prozent leicht auf 469 Millionen abnahm [29]. Mit der Erweiterung von UMTS durch *High Speed Packet Access* (HSPA) entstehen derzeit 3,5G-Netze mit Datenraten von bis zu 7,2 Mbit/s im Downlink und 1,45 Mbit/s im Uplink. Weiterhin wird im Rahmen des 3GPP mit dem *Long Term Evolution* (LTE)-Standard ein UMTS-Nachfolger entwickelt, der zukünftige 4G-Mobilfunknetze ermöglicht und ähnlich wie WLANs nach dem Standard 802.11n die MIMO-Technologie für weitaus höhere Datenraten z. B. im Bereich der Audio- und Videotelefonie oder als Rückkanal für mobiles interaktives Fernsehen nutzen wird.

1.2.4 Next Generation (Mobile) Networks

Im Zusammenhang mit Mobilfunknetzen der nächsten Generation tauchen immer wieder die Begriffe *Next Generation Networks* und *Next Generation Mobile Networks* auf, die für die Entwicklung des Internets eine besondere Bedeutung haben. Beide Begriffe werden nun kurz erläutert, um den Fokus dieser Dissertation klar davon abgrenzen zu können.

Mit dem steigenden Kostendruck im Telekommunikationsmarkt und dem damit einhergehenden Preisverfall bei Sprachdiensten mussten in klassischen drahtgebundenen Telekommunikationsnetzen neue Ansätze gefunden werden, um den Nutzern ein wirtschaftliches und effizientes Angebot von Telekommunikationsdiensten bereitstellen zu können. Unter dem Terminus *Next Generation Networks* (NGN) begann die ETSI in einer ersten NGN-Spezifikation in den 1990er Jahren mit der Migration vom leitungsvermittelten *Public Switched Telephone Network* (PSTN) über das *Integrated Services Digital Network* (ISDN) hin zu IP-basierten paketvermittelten Netzen, die traditionelle Telekommunikationsnetze wie Telefon- und Kabelnetze durch eine einheitliche paketvermittelte Netzinfrastruktur zunehmend ersetzen und als Plattform für das Angebot sämtlicher Dienste genutzt werden können. Mit dieser Migration werden große Einsparungen in der Netzinfrastruktur erwartet, da nicht mehr zwei parallel betriebene Netze zur Sprach- und Datenübertragung erforderlich sind. Die generelle Nutzung IP-basierter paketvermittelter Netze (z. B. zur VoIP-Telefonie) führt zur Abkopplung des Dienstangebots und des darunterliegenden Netzes. Sie führt aber auch dazu, dass beim Endkundenumsatz der Wertbeitrag der Netzbereitstellung zu Lasten des Dienstleistungsangebots abnimmt und sich die Wertschöpfungskette des Telekommunikationsmarktes verändert. Die Schwierigkeiten bei der Realisierung von NGNs liegen in der gleichzeitigen Verwendung von Diensten, die unterschiedliche Anforderungen an die Dienstgüte (Qualtity of Service (QoS)) stellen, so wie es z. B. bei der Übertragung von Sprache und Daten der Fall ist. Neben der Bereitstellung von breitbandigen Zugängen zum Internet liegen die großen Herausforderungen von NGNs also darin, eine umfassende Netzwerkarchitektur zu entwickeln, die Kontrollmechanismen bereitstellen, mit deren Hilfe die Netzwerkressourcen entsprechend den Anforderungen der Dienste und der Anzahl der Nutzer sinnvoll und gesteuert verwaltet werden können.

Unter dem Terminus *Next Generation Mobile Network* (NGMN) wird im Bereich der Erweiterung des Internets durch drahtlose Netzwerktechnologien besonders die Entwicklung der 4. Mobilfunkgeneration (4G) verstanden, die auf den bisherigen UMTS-Infrastrukturen aufsetzt und sie rasch sowie kostengünstig erweitert. Damit steht UMTS in den nächsten Jahren ein Generationswechsel zu dem vom 3GPP

standardisierten LTE bevor, bei dem die leistungsfähigere *Orthogonal Frequency Division Multiplex* (OFDM)-Modulation anstelle der derzeit verwendeten *Wideband CDMA* (WCDMA)-Modulation zum Einsatz kommen wird. Für mobile Endgeräte bedeutet NGMN, dass sie wie in WLANs permanent mit dem Internet verbunden sein können und ihnen deutlich höhere Bandbreiten mit bis zu 100 Mbit/s im Downlink und 50 Mbit/s im Uplink bei Latenzen von etwa 10 ms zur Verfügung stehen. Die Kompatibilität von NGMNs zu bereits vorhandenen Mobilfunknetzen soll dabei aber bewahrt bleiben.

Zusammenfassend stehen die Begriffe NGN und NGMN zum einen für die Konvergenz von unterschiedlichen Netzwerktechnologien, die z. B. das klassische drahtgebundene Telefonnetz und DSL als Zugangstechnologie zum Internet zu einem gemeinsamen IP-basierten Netzwerkanschluss verschmelzen. Zum anderen stehen sie für die Konvergenz von Diensten, die z. B. SMS, MMS, E-Mail und Instant Messaging zu neuen Diensten der Gruppenkommunikation und Gruppenkooperation verschmelzen. Vor allem stehen NGN und NGMN aber für die Abkopplung des Diensteangebots vom darunterliegenden Netzwerk, denn Dienste wie Telefonie, die vorher an das ISDN-Netz gebunden waren, werden nun in Form von VoIP-Diensten auf jeglichen IP-basierten Netzwerken unterstützt; egal ob dabei drahtgebundene Technologien wie DSL und Ethernet oder drahtlose Technologien wie WLAN und UMTS für den Zugang zum Internet genutzt werden.

1.2.5 Wireless Personal Area Networks

Gegenüber den bisher vorgestellten drahtgebundenen und drahtlosen Netzwerktechnologien, die vorrangig der Kommunikation mit dem Internet dienen und daher auf der Nutzung des Internet Protokolls basieren, entwickelten sich in den letzten Jahren eine Reihe weiterer Technologien, die eine spontane und direkte Vernetzung von Endgeräten in unserer unmittelbaren Umgebung und somit auch eine Verlängerung des Internets hinein in unser ganz persönliches Umfeld ermöglichen. Diese sogenannten *Wireless Personal Area Networks* (WPAN) wurden von der IEEE in der Standard-Familie 802.15 als Kurzstrecken-Funktechniken mit Reichweiten von 0,2 m bis 100 m spezifiziert. Sie ermöglichen eine Ad-hoc-Vernetzung von PDAs, Druckern, Laptops und Mobiltelefonen, die gegenüber der WLAN-Nutzung zwar nur niedrige Datenraten bereitstellt, aber durch den Einsatz geringerer Sendeleistungen viel Energie spart und so die Akkulaufzeit deutlich verlängert. Neben dem Austausch von kurzen Nachrichten, Sensorwerten, Visitenkarten und Kalendereinträgen zur Terminabstimmung wird teilweise auch die Übertragung von Dateien und Sprache unterstützt. Die derzeit bedeutendsten Vertreter von WPANs werden durch die Standards 802.15.1 und 802.15.4 unter den Namen Bluetooth bzw. ZigBee spezifiziert.

Bluetooth

Die Entwicklung der Bluetooth-Technologie zum Industriestandard begann im Jahre 1998 durch die von den Firmen Ericsson, Nokia, IBM, Toshiba und Intel gegründete *Bluetooth Special Interest Group* (SIG) [30] mit dem Ziel einen funkbasierten Ersatz für Kabelverbindungen zwischen Geräten zu realisieren, der das Kabelgewirr rund um einen Computerarbeitsplatz durch Funkperipherie für Tastatur, Maus und Drucker vermeidet. Mit der Bluetooth-Version 1.1 wurde im Jahre 2001 eine solide Kurzstrecken-Funktechnologie geschaffen, die je nach Geräteklasse bei einer Reichweite von 10, 20 oder 100 m eine Datenrate von 732,2 kbit/s erreichte. Mit den Versionen 2.0+EDR [31] im Jahre 2007 und der Version 3.0 [32] im Jahre 2009 wurde die Datenrate schrittweise über 2,1 Mbit/s auf 3 Mbit/s erhöht. Genau wie WLAN nutzt Bluetooth das ISM-Band im Bereich von 2,4 GHz zur Datenübertragung. Um Bluetooth jedoch robuster gegenüber Störungen durch WLANs (und Mikrowellengeräte) zu machen, wurde das genutzte Frequenzband in 79 Frequenzstufen mit 1 MHz Abstand eingeteilt und durch ein Frequenzsprungverfahren ergänzt, das bis zu 1600 mal pro Sekunde zwischen den Frequenzstufen wechselt. Die Hopping-Sequenz dieses Verfahrens wird dabei von einem sogenannten Master vorgegeben, der in einem Piconet die Kommunikation von bis zu 7 aktiven Slaves steuert und Sendeslots zur Kommunikation an die Slaves vergibt. Weitere 247 Slaves können in einem Parkmodus die Synchronisation zum Master halten und auf Anfrage im Piconet aktiviert werden. Damit übernimmt ein Master ähnlich zu einem Access Point in WLAN-Infrastrukturnetzen die Koordination eines Piconets und kann die Funksignalstärke zu den übrigen Netzknoten in Form eines *Received Signal Strength Indication* (RSSI)-Wertes bestimmen. Im Gegensatz zu WLAN kann ein Bluetooth-Gerät auch in mehreren Piconets angemeldet sein, allerdings nur in einem als Master fungieren. Durch diese Verbindung können Piconets zu einem sogenannten Scatternet verbunden werden, bei dem jedes Piconet durch eine eigene Hopping-Sequenz identifiziert wird. Zur Unterstützung des Ad-hoc-Charakters können Bluetooth-Geräte weitere in der Reichweite befindliche Bluetooth-Geräte durch eine Erkundungsnachricht (Inquiry) kontaktieren und eine Punkt-zu-Punkt-Verbindung mit ihnen aufbauen. Um flexibel auf neue Geräteanforderungen reagieren zu können, wurden für die Kommunikation zwischen Bluetooth-Geräten verschiedene Profile spezifiziert, die für bestimmte Anwendungsbereiche festgelegt sind. So können Profile, die einen *Synchronous Connection Oriented Link* (SCO) mit einer Datenrate von 64 kbit/s zur leitungsvermittelten synchronen Übertragung von Sprachdaten nutzen, für den Einsatz in Freisprecheinrichtungen und in Headsets für VoIP, aber auch für die HiFi-Musikwiedergabe eingesetzt werden. Profile, die dagegen einen *Asynchronous Connectionless Link* (ACL) für eine paketvermittelte asynchrone Datenübertragung nutzen, können in Eingabegeräten wie Maus, Tastatur und sonstigen Controllern, aber auch zur Synchronisation

von Kontakten und Musik oder auch im Bereich der Hausautomation als Funkschlüssel genutzt werden. Der Bluetooth-Standard bietet noch weitere Features, zu denen verschiedene Energiesparmodi wie Hold, Sniff und Park, sowie Sicherheitsmodi wie Pairing und Authentifizierung mit bis zu 16-stelliger *Personal Identification Number* (PIN) zur Verschlüsselung der transportierten Daten gehören, auf die im Rahmen dieser Arbeit aber nicht genauer eingegangen wird.

ZigBee

Neben der Bluetooth-Technologie gibt es mit ZigBee einen weiteren Vertreter eines WPANs, der auf Basis einer einfachen und preisgünstigen Funkverbindung eine spontane Kommunikation und Kooperation der in der Umgebung des Nutzers befindlichen Geräte unterstützt. ZigBee ist im Standard IEEE 802.15.4 als offenes Funknetz spezifiziert und wird durch die im Jahre 2002 gegründete und aus über 230 Unternehmen bestehende ZigBee-Allianz [33] weltweit weiterentwickelt. Zur Datenübertragung nutzt ZigBee sowohl das freie ISM-Frequenzband im Bereich von 2,4 GHz als auch das 915 MHz- und 868 MHz-Band, die Datenraten von 250, 40 bzw. 20 kbit/s ermöglichen [34]. Um technische Probleme der Interferenzen bei der Koexistenz von ZigBee, WLAN und Bluetooth im 2,4 GHz-Band zu vermeiden, werden ähnlich zu Bluetooth auch hier besondere Verfahren eingesetzt, die eine gegenseitige Verträglichkeit gewährleisten. Dazu setzt ZigBee zum einen das *Direct Sequence Spread Spectrum* (DSSS) als Frequenzspreizverfahren ein, zum anderen verwendet ZigBee 16 überlappungsfreie 2 MHz breite Kanäle, die jeweils 5 MHz auseinander liegen [35]. Im WLAN werden dagegen Kanäle mit einer Breite von jeweils 20 MHz eingesetzt, die bei einer überlappungsfreien Nutzung der Kanäle 1, 7 und 13 jeweils 30 MHz auseinander liegen und somit knapp 10 MHz große Lücken in der Frequenznutzung aufweisen. Da ZigBee das Potential hat, die Energie zu messen, die über die einzelnen Kanäle übertragen wird und diese Information auch höheren Protokollschichten mitteilt, wird ZigBee-Geräten die Möglichkeit gegeben, auf wenig benutzte Kanäle auszuweichen. Im Falle der Koexistenz mit WLAN sind dies besonders die gerade beschriebenen Lücken zwischen den WLAN-Kanälen 1, 7 und 13 [36].

Je nach Aufgabe werden ZigBee-Geräte in unterschiedliche Gerätearten eingeteilt. Bei der einfachsten Geräteart implementiert ein Gerät (z.B. ein Lichtschalter) als *Reduced Function Device* (RFD) nur einen Teil der ZigBee-Protokolle. Es meldet sich bei einem Router an und bildet mit ihm ein Netzwerk in Stern-Topologie. Die Geräteart der *Full Function Devices* (FFD) agiert als Router. Dadurch, dass sie sich ebenfalls bei einem existierenden Router anmelden können, bilden sie ein vermaschtes Netzwerk in Baum-Topologie, bei dem genau ein Router innerhalb eines WPANs

zusätzlich die Rolle des Koordinators übernimmt, die grundlegenden Parameter des WPANs vorgibt und so das Netzwerk verwaltet.

Mit Reichweiten von 10 m bis 100 m und einem noch geringeren Energieverbrauch als Bluetooth besitzt ZigBee eine große Bedeutung für die Anwendung in der Industrie- und Automatisierungstechnik zur Anlagensteuerung oder Güterüberwachung, die auf der Übertragung von Sensordaten basieren. Ebenso bietet ZigBee viele Einsatzmöglichkeiten in der Heim- und Gebäudeautomatisierung, wie z. B. dem Einsatz von wartungsfreien Funkschaltern und Funksensoren mit beschränkter Energieversorgung in schwer zugänglichen Bereichen oder auch in Bereichen der Unterhaltungselektronik und Computer-Peripherie.

1.2.6 Sensornetzwerke

Mit der andauernden Miniaturisierung eingebetteter Systeme und der Entwicklung energiesparsamerer Funktechnologien im Bereich der WPANs entstanden in den letzten Jahren immer kleinere und leistungsfähigere Geräte, die uns verstärkt in unserem alltäglichen Umfeld umgeben und sich zunehmend in Alltagsgegenständen verstecken. Diese sogenannten Sensorknoten, die aus einem Prozessor, einem Datenspeicher, teilweise mehreren Sensoren, einem Modul zur Funkkommunikation und einer Batterie zur Energieversorgung bestehen, kommen willkürlich zusammen und haben das technische Potential, vielfältige Umgebungsinformationen wahrzunehmen und sie in einem spontan gebildeten drahtlosen Kommunikationsnetz untereinander auszutauschen. Weiterhin können sie durch Aktoren auf spezielle Ereignisse in der Umgebung reagieren, ohne eine existierende Infrastruktur mit festen Basisstationen zu benötigen. Die vorrangige Multi-Hop-Kommunikation in diesen Ad-hoc-Netzen ist dabei nicht nur durch ein unvorhersagbares, dynamisches Netzwerkverhalten charakterisiert, sondern je nach Anwendungsfall auch von unterschiedlichen Netzwerkprotokollen, die sich für verschiedene Zwecke unterschiedlich gut eignen. Vor allem werden Sensornetze für den Austausch von kurzen Nachrichten eingesetzt, die geringe Datenraten benötigen und kaum QoS-Anforderungen haben. Mit dieser Art der Kommunikationsnetze kann eine flächendeckende Sensoranordnung z. B. zur Überwachung von Naturgebieten auf Schadstoffe oder als Frühwarnsystem für Waldbrände realisiert werden. Hierbei kommt gerade geographischen Routing-Verfahren eine besondere Bedeutung zu, da sich der Nutzer in vielen Anwendungsszenarien für Messdaten eines bestimmten geografischen Gebiets interessiert. Passive und praktisch unsichtbare Sensorknoten ermöglichen aber auch die Fernidentifikation von Dingen und können mit Hilfe umgebender Sensoren, die ihre Position kennen, für die präzise Ortsbestimmung und Ortung von Gegenständen eingesetzt werden. In derzeitigen Anwendungen werden Sensoren nicht nur in der Umgebung des Nutzers, sondern z. B. auch direkt

in Kleidung, Armbanduhren und Schmuckstücken eingebaut, um gesundheitlich relevante Parameter direkt am Körper zu messen und eine durchgängige Überwachung des Gesundheitszustandes zu ermöglichen. Sensoren, die als „Sinnesorgane“ smarter Dinge vielfältige Umweltparameter wahrnehmen, kommt damit vermehrt die Bedeutung zu, die Sinne des Nutzers zu erweitern, ihn mit Informationen zu versorgen und ihn in seinem Alltag sicherer und mächtiger zu machen.

Auch im Bereich der Heim- und Gebäudeautomation werden Sensornetzwerke zunehmend für dezentrale Überwachungs-, Steuer-, Regel- und Optimierungseinrichtungen eingesetzt, um nach vorgegebenen Parametern selbstständig auf sich wandelnde Umweltbedingungen zu reagieren. Dabei sorgt die Vernetzung von Sensoren, Aktoren, Bedienelementen, Verbrauchern und anderen technischen Einheiten im Gebäude dafür, dass ein Mehrwert für die Gerätenutzung entsteht und neue Bedienmöglichkeiten geschaffen werden. Im Heimbereich führt der anhaltende Trend der vernetzten Haushaltsgeräte (wie z. B. kleine elektrische Küchengeräte oder auch Großgeräte wie Kühlschrank, Waschmaschine, Herd und Geschirrspüler) zu neuen Möglichkeiten der Verbrauchsdatenerfassung von Wärme-, Wasser-, Gas- und Stromzählern. Darüber hinaus bieten Sensornetze neue Möglichkeiten für die Steuerung der Beleuchtung, erhöhte Sicherheit durch Überwachung von Fenster- und Türkontakten, eine zentrale Erfassung von Steuerungsvorgängen, die Steuerung der Multimedia-Geräte (z. B. Fernseher, Videorekorder, Tuner, zentraler Server) sowie die Fernüberwachung und Fernsteuerung. Durch den Einsatz sogenannter Residential Gateways [37], die als eine Art erweiterter Router die technische Schnittstelle zwischen Wohnung und Außenwelt darstellen, werden die drahtgebundenen und drahtlosen Netze und Bussysteme innerhalb einer Wohnumgebung sowohl miteinander als auch mit dem Internet verbunden. Damit übernimmt ein Residential Gateway nicht nur das teilweise vollständig autarke Management der unmittelbaren Umgebung, sondern auch das Remote Management zur Fernsteuerung der Wohnung über eine gesicherte Internet-Verbindung. Mit der Heim- und Gebäudeautomation ist ein bedeutender Gewinn an Wohnkomfort verbunden, der durch intelligente, teilweise autarke Steuerung der Geräte in unserer alltäglichen Umgebung eine messbare Energieverbrauchsreduktion, die Erhöhung der Sicherheit der Bewohner und die Möglichkeit der Überwachung mehrerer Wohnsitze bietet. Eine derartige Umgebung setzt eine durchgängige Vernetzung voraus, auf dem Protokoll-Stacks wie z. B. *Universal Plug and Play* (UPnP), *Open Services Gateway initiative* (OSGi), *Jini* oder der *European Installation Bus* (EIB) aufsetzen.

Im Rahmen der *European Installation Bus Association* (EIBA) schlossen sich im Jahre 1990 führende Hersteller zusammen um mit EIB einen Standard einzuführen, der die Kompatibilität und Interoperabilität der verschiedenen Geräte und Systeme unterschiedlicher Hersteller aus den Bereichen Klima & Lüftung und Hausgeräte gewährleistet. Mittlerweile steht die Erweiterung von EIB als EIB/KNX-Standard [38]

für eine ausgereifte und weltweit durchgesetzte intelligente Vernetzung moderner Haus- und Gebäudesystemtechnik gemäß EN 50090 und ISO/IEC 14543 und steuert gewerkeübergreifend und bedarfsgerecht Heizung, Beleuchtung, Jalousien, Belüftung und Sicherheitstechnik. Der EIB/KNX-Standard sieht die Trennung von Stromversorgung (230 V Wechselspannung) und Gerätesteuerung (max. 30 V Gleichspannung) vor und beschreibt, wie Sensoren und Aktoren bei der Installation in einem Haus durch Gruppenadressen miteinander verbunden werden müssen und miteinander kommunizieren. Der Datenaustausch erfolgt zwischen EIB/KNX-Geräten über Telegramme, wobei auch hier ein CSMA/CA-Verfahren zum Zugriff auf den gemeinsamen Bus genutzt und eine Datenrate von 9,6 kbit/s ermöglicht wird. Derzeitige Weiterentwicklungen der EIB/KNX-Technologie bestehen zum einen in der Realisierung spezieller EIB/KNX-Gateways, die die Daten mehrerer räumlich entfernter EIB/KNX-Busse über ein LAN tunneln und so die Steuerung der gesamten Gebäudeautomation über ein LAN ermöglichen. Zum anderen führen die Entwicklungen im Bereich des Ubiquitous Computing zunehmend zum Einsatz von drahtlosen Funktechnologien (wie z. B. ZigBee) im Bereich der Gebäudeautomation, wodurch EIB nicht mehr nur durch statische Sensoren in unserer Umgebung, sondern auch durch Sensoren in mobilen Alltagsgegenständen und sogar in Dingen, die wir direkt am Körper tragen, erweitert wird [39].

Mit der Entwicklung von LANs, WLANs, PANs und Sensornetzen breitet sich das Internet zunehmend bis in die Alltagsgegenstände hinein aus, die sich in unserer unmittelbaren Umgebung befinden und die wir zum Teil nur noch indirekt wahrnehmen. Mit Sensoren ausgestattet, nehmen sie schon heute vielfältige Umgebungsinformationen wahr, tauschen sie mit anderen Geräten in ihrer Nähe aus und reagieren durch Aktoren auf spezielle Ereignisse in der Umgebung. Damit diese Geräte zu smarten Alltagsgegenständen werden, die den Nutzer in seiner aktuellen Aktivität unterstützen, ist jedoch noch eine intelligente Zusammenarbeit der Geräte notwendig. Für die Alltagsgegenstände bedeutet es, dass sie wissen, wo sie sich gerade befinden, welche Dinge oder Personen in der Nähe sind und was in der Vergangenheit mit ihnen geschah. Es bedeutet aber auch, dass sie dieses Wissen mit anderen Dingen in ihrer Umgebung teilen und durch kontextbezogenes, situationsangepasstes und kooperatives Verhalten einen Plan zur Unterstützung der aktuellen Aktivität des Nutzers generieren und ausführen. Mit der Integration von smarten Gegenständen in die umfassende Struktur und die Dienste des Internets sowie der Assistenz der Nutzer durch Forschungsprojekte im Bereich der intelligenten Umgebungen steht dem heutigen Internet zu einem *Internet der Dinge* [3] noch ein drastischer Wandel mit großen Herausforderungen bevor.

1.3 Heterogene Netzwerke in intelligenten Umgebungen

Gerade im Bereich des *Ambient Assisted Living* (AAL) treffen verschiedene drahtgebundene und drahtlose Netzwerktechnologien aufeinander, die in Verknüpfung mit dem direkten Umfeld des Nutzers in sogenannten intelligenten Umgebungen durch spezifische Unterstützung im Alltag die Verbesserung der Lebensqualität von Menschen in verschiedenen Alters- bzw. Bevölkerungsgruppen zum Ziel haben [40]. Die Realisierung solcher Umgebungen ist technologisch aufwendig, da die einzelnen Komponenten sehr gut aufeinander und auf die Bedürfnisse ihrer Nutzer abgestimmt sein müssen. Die verwendeten Technologien sind dabei auf den Nutzer ausgerichtet und integrieren sich in dessen Lebensumfeld, ohne dass die Technik als solches in Erscheinung tritt. Unter dem Schlagwort AAL förderte das Bundesministerium für Bildung und Forschung im Rahmenprogramm „Mikrosysteme“ (2004-2009) [40] in den letzten Jahren die Entwicklung von Projekthäusern, Wohnanlagen und Modellwohnungen mit Fokus auf die Schwerpunkte Gesundheit und Home Care, Sicherheit und Privatsphäre, Versorgung und Hausarbeit sowie soziales Umfeld. Derzeit arbeiten national und international eine Vielzahl von Forschungsinstituten, Hochschulen, Unternehmen und Gremien an AAL-Lösungen, deren Ziele sich in die Bereiche gesundheitsorientierte Wohneinheiten [41][42], technikorientierte Wohneinheiten mit zentralisierter Mediennutzung [43] und Konstruktions- bzw. Design-orientierte Wohneinheiten [44][45] einordnen lassen. So wird z. B. im InHaus-Zentrum der Fraunhofer Gesellschaft [46] in Duisburg eine ganzheitliche Anwendungslösung für Wohn- und Nutzimmobilien (sog. Smart Home und Smart Building) entwickelt, die eine Steigerung der Effizienz beim Planen, Bauen und Betreiben der Immobilien und diversen Anwendungsprozessen im Bereich AAL fokussiert. Ein weiteres Forschungsprojekt im Bereich AAL stellt das von der *Deutschen Forschungsgemeinschaft* (DFG) geförderte Graduiertenkolleg *Multimodal Smart Appliance Ensembles for Mobile Applications* (MuSAMA) [47] der Universität Rostock dar, das umfangreiche Grundlagenforschung bezüglich intelligenter Umgebungen (sog. Smart Labs) in den Bereichen Smart Home und Smart Office betreibt.

Eng verwandt mit AAL ist der Begriff des *Smart Living*, wobei hier die Vernetzung und Automation von Haushalts- und Multimedia-Geräten im Vordergrund steht und weniger die Assistenzfunktion eines adaptiven Gesamtsystems. AAL und Smart Living ermöglichen beispielsweise Anwendungen im Bereich der kontextabhängigen und an die Gewohnheiten des Nutzers angepassten Beleuchtungs-, Raumtemperatur- und Musiksteuerung sowie Anwendungen im Bereich der Gebäudesicherheit. Weiterhin wird die Nutzung technischer Hilfsmittel zur Assistenz beim Lehren und Lernen unter Berücksichtigung aktueller Umgebungsbedingungen eingesetzt. Auch durch den zunehmenden demographischen Wandel, bei dem mitdenkende Systeme altersgerech-

te Hilfestellungen bzw. Vorschläge zur Problemlösung anbieten sollen oder sogar externe Dienstleister wie telemedizinische Zentren mit einbinden, wird die Entwicklung von AAL-Technologien vorangetrieben.

Der Themenbereich dieser Arbeit widmet sich der Interoperabilität der im Bereich AAL und Smart Living vorherschenden Heterogenität der vorhandenen Netzwerktechnologien. Als Teilbereich des MuSAMA-Projekts wird er im folgenden Abschnitt durch eine Einordnung in die AAL-Forschungslandschaft näher vorgestellt.

1.3.1 Vorstellung des MuSAMA-Projekts

Im Oktober 2006 startete an der Universität Rostock das von der Deutschen Forschungsgemeinschaft geförderte Graduiertenkolleg MuSAMA [47] mit einer voraussichtlichen Gesamtdauer von 9 Jahren. Thematisch sind an diesem Graduiertenkolleg die Bereiche Informatik und Elektrotechnik beteiligt. Gemeinsam wird durch die interdisziplinäre Zusammenarbeit der Lehrstühle „Computergraphik", „Softwaretechnik“, „Datenbank- und Informationssysteme“, „Mobile Multimediale Informationssysteme“, „Mikroelektronik und Datentechnik“, „Modellierung und Simulation“, „Informations- und Kommunikationsdienste“ sowie „Rechnerarchitektur“ mit 10 Hochschullehrern, 14 Stipendiaten und 13 Kollegiaten an der Entwicklung von Konzepten und Verfahren für intelligente heterogene Umgebungen geforscht. Die bisher entwickelten Konzepte, sowie die Konzepte der vorliegenden Arbeit, konnten sowohl im Umfeld des MuSAMA-Projekts als auch im Rahmen von Publikationen bei internationalen, wissenschaftlichen Konferenzen ausführlich diskutiert und reflektiert werden.

Das MuSAMA-Projekt geht davon aus, dass unsere zukünftigen Umgebungen von unabhängigen, intelligenten Alltagsgegenständen geprägt sein werden. Eine lokale Ansammlung dieser stationären bzw. mobilen Geräte wird dabei als Ensemble angesehen, in dem Geräte spontan und sinnvoll miteinander kooperieren und den Nutzer zielgerichtet bei der Gestaltung, Organisation und Durchführung seiner Aufgaben im Alltag unterstützen. Dazu müssen sich Ensembles der Ziele und Bedürfnisse ihrer Nutzer bewusst sein und die Unterstützung in die jeweilige Aktivität des Nutzers einbetten, ohne ihn von seiner Primäraktivität abzulenken. Für eine derartige intelligente, spontane und autonome Kooperation der Geräte eines Ensembles werden Verfahren benötigt, mit denen sie selbstständig untereinander aushandeln können, welche Assistenz der Nutzer benötigt und wie diese Assistenzleistung kooperativ erbracht werden kann. Die unvorhersehbare Struktur eines Ensembles ist dabei eine wesentliche Herausforderung, die einen Rückgriff auf vordefinierte Reaktionsschemata

verhindert und die Entwicklung von Modellen für verteilte Abstimmungsmechanismen und Verfahren zur spontanen Kooperation erfordert. Dazu müssen sowohl die Intentionen des im Ensemble handelnden Nutzers erkannt, als auch Strategien für geeignete Aktionen bzw. Reaktionen bestimmt und gemeinschaftlich ausgeführt werden. Ensemble sollen dabei ohne globale Informationen handlungsfähig sein und keine koordinierenden bzw. konfigurierenden Eingriffe durch den Nutzer oder den Entwickler erfordern. Auf diese Weise sollen in den Bereichen Smart Home und Smart Office aus aktuellen Sensordaten Interaktionsereignisse und lokale Vermutungen über die Präferenzen und Handlungsziele der Nutzer zu einer sinnvollen Hypothese des aktuellen Unterstützungsbedarfs abgeleitet und auf deren Basis die Realisierung der erforderlichen Assistenz durch das Ensemble ermöglicht werden.

Bei diesen Herausforderungen ergeben sich im MuSAMA-Projekt die in Abbildung 1.4 dargestellten Forschungsbereiche. Der Bereich „Kontexterkennung und –analyse“

Abbildung 1.4 Fachliche Struktur des MuSAMA-Projekts

nutzt verteilte und vernetzte Sensorik für Modelle zur Bestimmung des aktuellen Kontextes, insbesondere die räumliche Konfiguration von Nutzern und Geräten in der aktuellen Umgebung. Im Bereich „Multimodale Interaktion und Visualisierung“ wird an der Visualisierung von Informationen in verteilten Infrastrukturen und auf ubiquitären Displays geforscht. Ein weiterer Bereich befasst sich mit der „Intentionserkennung und Strategieentwicklung“ um die Intention des Nutzers im Ensemble auf Basis von Kontextdaten, Nutzerpräferenzen und typischen Handlungsabläufen zu erkennen und eine Strategie für das Erreichen des Nutzerziels bereitzustellen.

Im Bereich „Datenhaltung, Ressourcen- und Infrastrukturmanagement“ ist der Lehrstuhl für Rechnerarchitektur sowohl durch die vorliegende Arbeit zum Thema „Allge-

genwärtige Kommunikation heterogener Systeme in Smart Ensembles“ als auch durch die Arbeit von Herrn Raphael Zender zum Thema „Dienstbasierte Gerätekommunikation unter Berücksichtigung von Kontextinformationen“ vertreten, um die Mechanismen der Kommunikation in dynamischen heterogenen Netzwerken zu untersuchen und geeignete Modelle für die Interoperabilität und eine effiziente Kommunikation in intelligenten heterogenen Ensembles zu realisieren.

1.3.2 Einordnung und Ziele der vorliegenden Arbeit

Bisher wurde gezeigt, dass mit der Entwicklung des Internets immer neue Möglichkeiten der globalen Kommunikation und Kollaboration bereitstehen, die mit der Entwicklung drahtloser WLANs, WPANs und Mobilfunknetze sowie immer kleiner werdender Geräte, wie z. B. Laptops, PDAs und Mobiltelefone schon heute einen weltweiten orts- und zeitunabhängigen Zugriff auf Informationen erlauben. Mit der andauernden Entwicklung von noch kleineren und sparsameren Geräten bzw. Sensoren dringt das Internet in Form eines *Internets der Dinge* bis in unsere unmittelbare Umgebung vor und stellt so als Verlängerung des Internets bis in die letzten Alltagsgegenstände einen wesentlichen Katalysator heutiger und zukünftiger technologischer Entwicklungen im Bereich intelligenter Umgebungen dar.

Im Gegensatz zu klassischen Netzwerkarchitekturen fokussieren intelligente Umgebungen, so wie sie auch im MuSAMA-Projekt eingesetzt werden, eine Informationsverarbeitung, die ubiquitär über Technologiegrenzen hinweg und mit Fokus auf das kooperative Verhalten der Geräte zur proaktiven Unterstützung des Nutzers in seiner Umgebung durchgeführt wird. Dabei sind aber gerade die in intelligenten Umgebungen anzutreffenden Netzwerke in unterschiedlichen Bereichen von Heterogenität gekennzeichnet. Zum einen unterscheiden sich die in einem Ensemble befindlichen Geräte in den ihnen zur Verfügung stehenden Ressourcen (z. B. CPU, Speicher und Energieversorgung). Zum anderen nutzen sie mit LAN, WLAN, UMTS, Bluetooth und ZigBee je nach Einsatzzweck unterschiedliche Kommunikationstechnologien, die sich in ihren Kommunikationsprotokollen und den Parametern Reichweite, Latenz, Datenrate und Energieverbrauch teilweise deutlich unterscheiden sowie mit der Nutzung technologieabhängiger IP- bzw. nicht-IP-basierter Kommunikationsprotokolle zusätzlich zur Heterogenität beitragen.

Aus Sicht des Themas dieser Arbeit stellt gerade die bisher fehlende bzw. unzureichende Interoperabilität der von den Geräten verwendeten Kommunikationstechnologien und den dabei eingesetzten Protokollen den Schwerpunkt dieser Arbeit im Bereich „Datenhaltung, Ressourcen- und Infrastrukturmanagement“ des MuSAMA-

Projekts dar. In der vorliegenden Arbeit werden daher die folgenden Kernziele verfolgt:

- Basierend auf Untersuchungen der Kommunikationsmechanismen dynamischer heterogener Netzwerke ist ein geeignetes allgemeingültiges Konzept zur Unterstützung der Interoperabilität verschiedener IP- und nicht-IP-basierter Netzwerktechnologien zu entwickeln. Dabei ist zu beachten, dass die Geräte eines Ensembles teilweise über Netzwerkschnittstellen unterschiedlicher Technologien verfügen, die weder eine durchgängige IP-basierte Adressierung noch die gleichen Kommunikationsprotokolle unterstützen. In dem zu entwickelnden Konzept müssen die verschiedenen Kommunikationstechnologien derart zusammengeführt werden, dass sie sich in ihren unterschiedlichen Eigenschaften (z. B. Reichweite, Bandbreite, Energieverbrauch, Adressierungsart und Kommunikationsprotokoll) gegenseitig ergänzen und eine effiziente, heterogene Kommunikation zwischen beliebigen Geräten sowohl innerhalb eines Ensembles als auch Ensemble-übergreifend ermöglichen. Die Mechanismen zur Realisierung der Interoperabilität der unterschiedlichen Netzwerktechnologien müssen durch spezielle Geräte im Netzwerk bereitgestellt werden, für die übrigen Geräte sollen sie jedoch transparent sein.
- Ausgehend von dem allgemeingültigen Konzept zur Unterstützung der Interoperabilität verschiedener IP- und nicht-IP-basierter Netzwerktechnologien ist eine einfach erweiterbare Referenzarchitektur als Basis eines interoperablen und effizienten Kommunikationssystems zu entwickeln und zu evaluieren, die die Vermittlerfunktionalität zwischen den Technologien im Ensemble übernimmt und so die Heterogenität aktueller pervasiver Umgebungen systematisch überwindet. Weiterhin ist es notwendig, das sie die transparente Einbindung neuer drahtgebundener und drahtloser Netzwerktechnologien jeglicher Art ermöglicht, um den sogenannten *Grand Challenges der technischen Informatik* [18] Rechnung tragen zu können, von denen eine besonders hohe wirtschaftliche Bedeutung im Anwendungsgebiet der *zukünftigen Kommunikationsnetze* und der *omnipräsenten Informationsverarbeitung* erwartet wird. Damit wird sowohl die Interoperabilität und Kooperation von Sensorsystemen als auch deren Integration in das allgemein verfügbare Internet gefördert und der verantwortungsbewusste Umgang mit Energie berücksichtigt.

1.4 Aufbau dieser Arbeit

Nachdem in diesem Kapitel die vorliegende Arbeit thematisch in den Rahmen des MuSAMA-Projekts eingeordnet und die Ziele formuliert wurden, stellt Kapitel 2 die Grundlagen sowie den Stand der Forschung zur Interoperabilität unterschiedlicher Netzwerktechnologien dar, die für das Verständnis der weiteren Arbeit von Bedeutung sind. Darauf aufbauend entwickelt Kapitel 3 das Konzept und damit die theoretischen Grundlagen einer allgegenwärtigen Kommunikation in intelligenten Umgebungen in Form einer Referenzarchitektur zur Kommunikation in heterogenen Netzwerken. Dabei wird mit einer theoretischen Betrachtung zuerst ein genauerer Blick auf die aktuelle Kommunikation in heterogenen Netzen geworfen und besonders auf horizontale und vertikale Netzwerkstrukturen sowie deren Kombination zu einem heterogenen Ensemble eingegangen. Danach werden zwei Organisationsformen für Gateways vorgestellt, die als Vermittler zwischen den verschiedenen Netzwerktechnologien eingesetzt werden können. Anschließend wird ein Ansatz zur Servicebasierten Kommunikation vorgeschlagen, der in Kombination mit einem adressbasierten Kommunikationsansatz für heterogene Netzwerke, der Mobilität von Endgeräten und deren Diensten Rechnung trägt. Danach wird die Referenzarchitektur für den Aufbau der Software eines General Purpose Access Points definiert, der das mit dieser Arbeit entwickelte Kommunikationskonzept für heterogene Netzwerke umsetzt. Das Kapitel 4 widmet sich der Evaluation der entstandenen Referenzarchitektur, bevor Kapitel 5 die Ergebnisse der Arbeit abschließend zusammenfasst und Anregungen für die zukünftige Entwicklung in diesem Forschungsbereich liefert.

Kapitel 2

Struktur und Funktionsweise heterogener Netzwerke

Nachdem diese Arbeit im vorherigen Kapitel motiviert und thematisch in den Bereich „Datenhaltung, Ressourcen- und Infrastrukturmanagement“ des MuSAMA-Projekts eingeordnet wurde, werden nun einige Grundlagen vorgestellt, die für das im nächsten Kapitel folgende Konzept einer allgegenwärtigen Kommunikation in heterogenen Netzwerken von tragender Bedeutung sind. Dazu stellt dieses Kapitel einleitend den Aufbau heterogener Netzwerke vor und geht genauer auf die Grundlagen der Kommunikation in den IP-basierten und nicht-IP-basierten Netzwerkbereichen ein. Dabei werden Adressierungs- und Kommunikationsformen beider Bereiche vorgestellt, bevor näher auf den aktuellen Stand der Forschung im Themenbereich der Interoperabilität unterschiedlicher IP-basierter und nicht-IP-basierter Netzwerktechnologien eingegangen wird.

2.1 Struktur heterogener Netzwerke

Mit den im Abschnitt 1.2 beschriebenen Netzwerktechnologien umgeben uns derzeit eine Vielzahl unterschiedlicher drahtgebundener und drahtloser Kommunikationstechnologien, die sich in der Art der Datenübertragung (elektrisch, optisch oder elektromagnetisch), ihrer Netzwerkstruktur (Infrastruktur-, Ad-hoc- oder Mesh-Netz), ihrer Reichweite und den ihnen zur Verfügung stehenden Datenraten unterscheiden. Allein gesehen stellt jede Technologie ein homogenes Netzwerk dar, in dem Geräte, die über gleiche Netzwerkschnittstellen verfügen, miteinander kommunizieren können. In der Gesamtheit führt der Einsatz der Technologien zu einem heterogenen Netzwerk, das wie in Abbildung 2.1 dargestellt, aus unterschiedlichen homogenen Netzwerken besteht, die über Gateways und Router an das öffentliche Internet angeschlossen sind und dieses bis in unsere unmittelbare Umgebung erweitern.

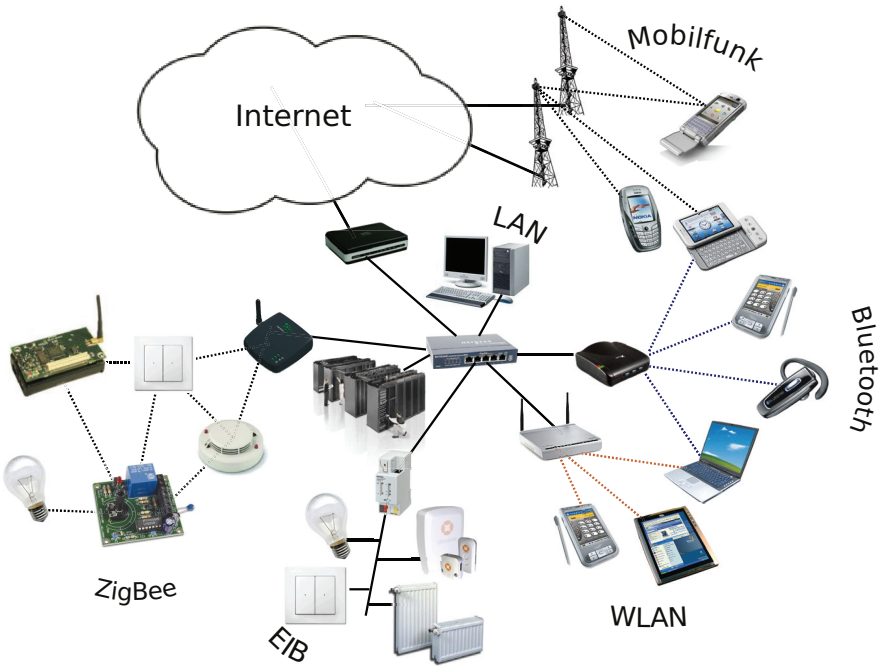

Abbildung 2.1 Aufbau eines heterogenen Netzwerks

In einem heterogenen Netzwerk stellen LANs, die auf dem Ethernet-Standard basieren, eine klassische Erweiterung des Internets dar. Durch eine strukturierte Verkabelung werden PCs, Workstations und Server mit Hilfe aktiver Switche zu einem lokalen Netzwerk mit Baumtopologie verbunden. Somit wird es den Geräten ermöglicht, mit allen anderen Geräten in ihrer Broadcast-Umgebung direkt per *Media Access Control* (MAC)-Adresse zu kommunizieren. Durch Nutzung einer IP-basierten Adressierung und Einsatz eines Internet-Gateway-Routers kann die Broadcast-Domäne verlassen und so ein Zugang zum Internet ermöglicht werden. Dazu besitzt der Router eine IP-Adresse im öffentlichen Internet sowie eine IP-Adresse im privaten LAN und übernimmt die Vermittlerrolle zwischen den unterschiedlichen Adressbereichen.

Gegenüber LANs können WLANs in unterschiedlichen Modi betrieben werden, die die Topologie des Netzes maßgeblich beeinflussen. So können WLANs, z. B. wie in Abbildung 2.1 dargestellt, in einem Infrastrukturmodus arbeiten, in dem ein Access Point die Koordination der Kommunikation zwischen den überwiegend mobilen Geräten übernimmt und den Zugriff auf das Übertragungsmedium steuert. Mit WLAN ausgestattete Geräte können aber auch in einem Ad-hoc- bzw. Mesh-Modus arbeiten, in dem gleichberechtigte Endgeräte zu einem vermaschten Netzwerk verbunden werden. Der Aufbau und die Konfiguration des Netzes werden dabei selbstständig durch die Endgeräte koordiniert. Da die Weiterleitung der Daten im Standard 802.11 nicht vorgesehen ist, sind hier spezielle Ad-hoc-Routingprotokolle nötig, um

einen Überblick über die Netzwerktopologie und darauf aufbauende Entscheidungen für die Weiterleitung der Nutzdaten ermöglichen zu können. In beiden Modi ist die Kopplung mit einem drahtgebundenen LAN vorgesehen. Dazu übernehmen ein AP im Infrastrukturmodus bzw. ein ausgewähltes Endgerät im Mesh-Modus, die jeweils über eine LAN- und eine WLAN-Schnittstelle verfügen, die Datenvermittlung zwischen den beiden Netzen. Auf diese Weise wird das Internet über das LAN bis in das WLAN erweitert und eine IP-basierte Kommunikation zwischen den Teilnehmern dieser Netze ermöglicht.

Als datenoptimierte Netze standardisiert, ermöglichen auch Mobilfunknetze der 3. und 4. Generation den Einsatz von IP-Adressen vom Internet bis hin zum mobilen Gerät. Ähnlich dem Infrastrukturmodus in WLANs ermöglicht hier ein aus mehreren Mobilfunkstationen bestehendes Zugangsnetz die Koordination der Datenübertragung zwischen den mobilen Endgeräten. Zusätzlich realisiert ein Mobilvermittlungsnetz die Übertragung und Vermittlung der Signale zwischen den Mobilfunkstationen und ermöglicht ein Handover der Verbindungen beim Wechsel der aktuell zuständigen Mobilfunkstation.

Neben diesen IP-basierten Netzwerken, die das Internet durch die Nutzung von LAN, WLAN und Mobilfunk bis hin zu unseren mobilen Computern erweitern, liegen die Herausforderungen zukünftiger heterogener Netzwerke in der Integration und Interoperabilität energiesparsamer Netzwerke, die eine spontane und direkte Vernetzung von Endgeräten in unserem ganz persönlichen Umfeld ermöglichen und auf Grund ihrer Struktur keine IP-Adressen benötigen. Zu diesen Netzwerktechnologien gehören Bluetooth und ZigBee, die als WPANs zwar nur geringe Datenraten bereitstellen, aber durch den Einsatz geringer Sendeleistungen energiesparsam arbeiten und die Akkulaufzeit der Geräte deutlich verlängern.

Die Kommunikation basiert in Bluetooth-Netzen auf einem Master-Slave-Prinzip, bei dem eine Gruppe von bis zu 8 aktiven Geräten ein sogenanntes Piconet bildet. In diesem Piconet übernimmt ein beliebiges Gerät die Rolle des Masters und gibt die Hopping-Sequenz des Netzes vor. Die Kommunikation zwischen den Slaves wird dann wie bei APs im WLAN über den Master durchgeführt. Ein besonderes Feature von Bluetooth ist, dass es eine Ad-hoc-Kommunikation von nahe gelegenen Geräten ermöglicht, die von der Nutzung der IP-Adressen unabhängig ist. Dadurch entfallen Aufgaben höherer Protokollschichten, wie die Vergabe von IP-Adressen und Netzwerkeinstellungen (wie z. B. Default Gateway und Netzwerkmaske).

Mit der stärkeren Fokussierung auf Sensornetze kombiniert ZigBee verschiedene Formen der Netztopologie. Da einfache Geräte, wie die Steuerung einer Lampe, nur einen Teil der ZigBee-Protokolle unterstützen müssen, melden sie sich an einem ZigBee-Router ihrer Wahl an und bilden mit ihm ein Netzwerk in Stern-Topologie. Router

melden sich wiederum jeweils an einem bereits existierenden Router im ZigBee-Netz an und bilden so ein Netzwerk in Baum-Topologie, wobei durch Ausnutzung von Abkürzungen ein vermaschtes Netzwerk entsteht, in dem genau ein Router zusätzlich die Rolle des Koordinators übernimmt und grundlegende Parameter des PANs vorgibt. Typischerweise wird ein Router als Koordinator vorkonfiguriert oder der erste aktive Router wird automatisch zum Koordinator. Bei Ausfall des Koordinators muss das Netz erneut aufgebaut werden und einen Koordinator auswählen [48].

Gerade im Bereich der Vernetzung und Automation von Hausgeräten werden die in unserer Umgebung befindlichen Sensoren zur Messung von Temperatur, Helligkeit und Position sowie Aktoren zur Steuerung der Beleuchtung, der Heizungs-, Klima- oder Schließanlage über drahtgebundene Bussysteme miteinander verbunden. Eines der bedeutendsten Bussysteme stellt der EIB dar, dessen kleinste funktionsfähige Einheit eine sogenannte Linie ist. Logisch ist eine Linie durch eine Bustopologie gekennzeichnet; hinsichtlich der Leitungsverlegung kann sie aber als Linien-, Stern-, oder Baumtopologie realisiert werden, nur eine Ringtopologie ist nicht vorgesehen. An dieser Linie können insgesamt bis zu 64 Sensoren und Aktoren teilnehmen, die von einer Kleinspannungsversorgung (30 V DC) über die Busleitung gespeist werden. Um größere EIB-Netze zu realisieren, können bis zu 15 Linien durch Linienkoppler über Hauptlinien zu einem Bereich zusammengefasst werden, von denen wiederum bis zu 15 Bereiche durch Bereichskoppler über Bereichslinien zu einem größeren EIB-Netz kombiniert werden können. Weiterhin ermöglichen Busankoppler die Kombination mit einem LAN und so die Programmierung und Steuerung eines EIB-Netzes durch z. B. einen PC. Auch eine Kombination zweier räumlich entfernter EIB-Busse ist mit diesen Kopplern über das LAN bzw. das Internet möglich.

Die Nutzung verschiedener Netzwerktechnologien führt dazu, dass sich das Internet über LANs, WLANs und Mobilfunknetze bis hin zu unseren mobilen Geräten ausweitet und sich zu einem IP-basierten Kernnetz entwickelt. Innerhalb unseres persönlichen Umfelds umgeben uns mit Bluetooth, ZigBee und EIB jedoch vorrangig Technologien, die durch eine spontane und direkte Vernetzung kleinerer Endgeräte gekennzeichnet sind und auf Grund ihrer Struktur keine IP-basierte Kommunikation benötigen. Dadurch ergibt sich ein Zwiespalt zwischen IP-basierten und nicht-IP-basierten Netzen, der die Interoperabilität unterschiedlicher Netzwerktechnologien und eine mit dieser Arbeit angestrebte allgegenwärtige Kommunikation zwischen beliebigen Teilnehmern im heterogenen Netzwerk bisher erschwert bzw. verhindert. Die folgenden Abschnitte geben daher einen tieferen Einblick in die Grundlagen der unterschiedlichen Adressierung und Kommunikation beider Netzwerkbereiche, bevor der Abschnitt 2.4 genauer auf den aktuellen Stand der Forschung zur Interoperabilität unterschiedlicher IP-basierter und nicht-IP-basierter Technologien eingeht.

2.2 Adressierung und Kommunikation in IP-basierten Netzen

Mit dem im Jahre 1981 von der *International Organization for Standardization* (ISO) entwickelten und 1984 standardisierten *Open Systems Interconnection Reference Model* (OSI-Referenzmodell) [21] wurde ein offenes Schichtenmodell realisiert, das vereinfachte Regeln für die Kommunikation auf dem Gebiet der Rechnersysteme festlegt. Das OSI-Modell dient als Grundlage einer Reihe herstellerunabhängiger Netzwerkprotokolle und als Entwicklungsvorschrift für Hardware und Software. Es sichert somit die Kooperation von Teilkomponenten verschiedener Hersteller. Auf Grund der unterschiedlichen Anforderungen, die von der elektronischen Übertragung der Signale, über eine geregelte Ablaufreihenfolge in der Kommunikation, bis hin zu abstrakteren Aufgaben innerhalb kommunizierender Anwendungen reichen, ist das OSI-Modell in 7 aufeinander aufbauende Schichten aufgeteilt, die jeweils eine Instanz der Anforderungen umsetzen. Dabei bedient sich eine Instanz den Diensten einer unmittelbar darunterliegenden Instanz und stellt einer darüber liegenden Instanz eigene Dienste zur Verfügung. Abbildung 2.2 stellt die Schichten des OSI-Modells sowie ausgewählte Instanzen genauer dar.

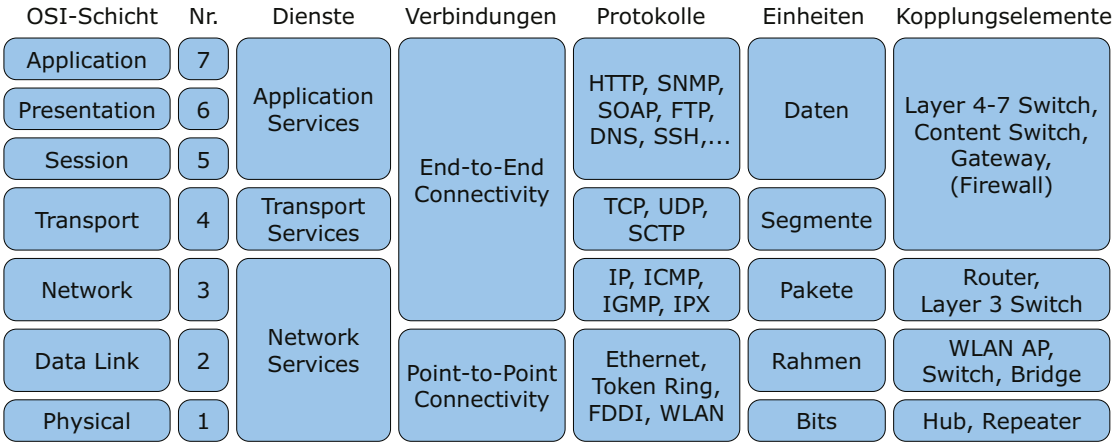

Abbildung 2.2 Schichten und Instanzen des OSI-Referenzmodells

Durch Abstraktion der spezifischen Funktion können nicht nur Implementierungsdetails nach außen verborgen, sondern auch einzelne Schichten von verschiedenen Herstellern implementiert und unabhängig von den anderen Schichten ausgetauscht werden. Zur Optimierung der Kommunikation können aber auch mehrere Schichten zu einer Schicht zusammengefasst werden.

Im Bereich der IP-basierten Netzwerke werden derzeit etwa 500 Protokolle, die die Basis für die Kommunikation im Internet bilden und dementsprechend eine große Bedeutung für das Internet haben, zu der sogenannten Internetprotokollfamilie zusam-

mengefasst [49]. Die Protokolle dieser Protokollfamilie nutzen das in Abbildung 2.3 dargestellte TCP/IP-Referenzmodell zur Gliederung der Kommunikationsaufgaben in funktionale Ebenen. Das TCP/IP-Referenzmodell wurde im RFC 1122 [22] definiert und fasst die Schichten des OSI-Modells in vier aufeinander aufbauenden Schichten zu einem Protokollstapel zusammen, der den Aufbau und das Zusammenwirken der Protokolle der Internetprotokollfamilie beschreibt. Die Internet-Protokolle sind dafür zuständig, dass Datenpakete über mehrere Punkt-zu-Punkt-Verbindungen weitergeleitet und auf dieser Basis Verbindungen zwischen entfernten Netzwerkteilnehmern hergestellt werden. Die dazu notwendige Datenübertragungstechnik sowie der Zugriff auf ein Übertragungsmedium werden jedoch nicht definiert.

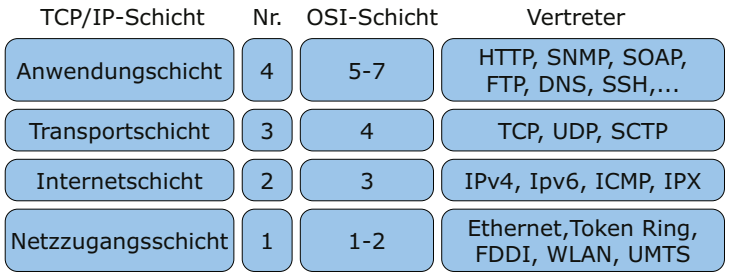

Abbildung 2.3 Aufbau des TCP/IP-Referenzmodells

Die **Netzzugangsschicht** bildet die unterste Ebene des TCP/IP-Referenzmodells und entspricht sowohl dem Physical Layer als auch dem Data Link Layer des OSI-Modells. Sie dient als Platzhalter für verschiedene Technologien zur Datenübertragung auf Basis von Punkt-zu-Punkt-Verbindungen und enthält daher keine Protokolle der Internetprotokollfamilie. Mit den Technologien Ethernet, FDDI, Token Ring und WLAN können hier Hubs, Repeater, Switche und APs eingesetzt werden, um verschiedene Netzwerksegmente miteinander zu verbinden.

Die **Internetschicht** des TCP/IP-Modells entspricht dem Network Layer des OSI-Modells. Damit ist sie für die Wegewahl und die Weitervermittlung (das sogenannte Routing) von Datenpaketen verantwortlich. Indem Router eingesetzt werden, die für jedes empfangene Datenpaket das nächste Zwischenziel ermitteln und das Paket dorthin weiterleiten, ermöglicht die Internetschicht eine unzuverlässige, verbindungslose Paketvermittlung für die höheren Schichten.

Als dritte Schicht des TCP/IP-Referenzmodells übernimmt die **Transportschicht** die Aufgaben der gleichnamigen Schicht des OSI-Referenzmodells und stellt mit Hilfe der Protokolle *Transmission Control Protocol* (TCP), *User Datagram Protocol* (UDP) und *Stream Control Transmission Protocol* (SCTP) eine Ende-zu-Ende-Kommunikation her. Je nach Art der Anforderungen können dabei mit UDP, einem

minimalen, verbindungslosen Protokoll; mit TCP, einem zuverlässigen, verbindungsorientierten Protokoll; oder mit SCTP, einem zuverlässigen, verbindungsorientierten Protokoll, das mehrere Verbindungen gleichzeitig nutzt, recht unterschiedliche Transportprotokolle zur Datenübertragung eingesetzt werden.

Die **Anwendungsschicht** übernimmt als oberste Schicht des TCP/IP-Referenzmodells die Aufgaben des Session, Presentation und Application Layers des OSI-Modells und umfasst somit alle Protokolle, die die Netzwerkinfrastruktur für den Austausch anwendungsspezifischer Daten nutzen. In diese Schicht lassen sich zum Beispiel die Protokolle HTTP, FTP, DNS, SSH und SOAP einordnen.

Gegenüber dem OSI-Referenzmodell legt das TCP/IP-Modell die Funktionen und Dienste der einzelnen Schichten nicht zu genau fest und erlaubt es so, einzelne Schichten zu umgehen. Auf diese Weise können untere Schichten direkt und effizienter von höheren Schichten benutzt werden. Bei derartigen Optimierungen können jedoch eine Vielzahl eigenständiger Netzwerkprotokolle entstehen, die nur dann zusammenarbeiten, wenn auf beiden Seiten der Kommunikation das gleiche Protokoll unterstützt wird.

2.2.1 Adressierung

In IP-basierten Netzen wird für die Adressierung eines Kommunikationspartners sowohl auf die Netzzugangsschicht als auch auf die Internetschicht zugegriffen. Auf der Netzzugangsschicht werden auf Basis von MAC-Adressen [50], die jeweils einem Netzwerkgerät vom Hersteller als Hardware-Adresse zugeordnet sind und somit der eindeutigen Identifizierung des Gerätes in einem Rechnernetz dient, Punkt-zu-Punkt-Verbindungen (sogenannte Hops) zwischen Geräten im selben Netzwerksegment ermöglicht. Im Falle der Verwendung der Ethernet-Technologie werden die zwischen den Kommunikationspartnern zu übermittelnden Daten in mehrere kleine Pakete aufgeteilt und in sogenannten Frames übertragen [20].

In der Internetschicht werden Router eingesetzt, die für jedes empfangene Datenpaket das nächste Zwischenziel ermitteln und das Paket dorthin weiterleiten. Hierbei werden statt Punkt-zu-Punkt-Verbindungen, nun Ende-zu-Ende-Verbindungen realisiert, die eine zusätzliche Quell- und Ziel-Adresse in Form einer IP-Adresse benötigen. Die IP-Adressen, die jede Netzwerkschnittstelle im Netzwerk unterscheidbar machen, geben dabei die Anfangs- und Ende-Adresse der Ende-zu-Ende-Verbindung an und ermöglichen ein Überschreiten der Netzwerkgrenzen durch eine technologie- und netzübergreifende Adressierung. Als Ziel-Adresse kann hierbei auch eine Broadcast- oder Multicast-Adresse eingesetzt werden, um ein Paket an mehrere Stationen weiterzuleiten. Die in der Netzzugangsschicht verwendeten MAC-Adressen geben dagegen nur

die Anfangs- und Ende-Adresse des jeweiligen Hops in Richtung der Ziel-IP-Adresse an. Der Kern der Internetschicht besteht somit aus dem *Internet Protocol* (IP) [6], das derzeit vorrangig in der Version 4 eingesetzt und zunehmend durch die Version 6 ersetzt wird.

2.2.2 Adressbasierte Kommunikation

In weltweiten IP-basierten Netzwerken wird die Ende-zu-Ende-Kommunikation zwischen beliebigen Netzwerkteilnehmern durch die Transportschicht des TCP/IP-Modells realisiert, die auf Basis der Netzwerkschicht und den damit verfügbaren ungesicherten, paketorientierten IP-Datagrammen eine sichere, verbindungsorientierte Kommunikation für beliebige Datengrößen ermöglicht. Die Transportschicht gilt als übliche Anwendungsschnittstelle und bietet der Anwendungsschicht einen einheitlichen Zugriff auf die Nutzung von Kommunikationsverbindungen zu anderen IP-basierten Teilnehmern. Zu ihren Aufgaben zählt neben der Segmentierung von Datenpaketen, dem Multiplexing von Datenströmen sowie der Fehlersicherung und Fehlerkorrektur aber auch die Stauvermeidung durch eine Flusssteuerung, die für einen kontinuierlichen Datenfluss sorgt. Je nach Art der Anforderung an die Kommunikationsverbindung, können dabei die Protokolle UDP, TCP oder auch SCTP eingesetzt werden. Alle drei Protokolle nutzen eine sogenannte Port-Struktur, die ein Multiplexing erlaubt und jede Netzwerkverbindung einer dafür zuständigen Anwendung zuordnet. Auf diese Weise können auf einem Netzwerkgerät mehrere Netzwerkverbindungen durch unterschiedliche Anwendungen gleichzeitig genutzt werden, die sich durch den von ihnen reservierten Port unterscheiden und so die Übergabe der Daten an die richtige Anwendung sicherstellen. Mit Hilfe eines Ziel-Ports wird eine bestimmte Anwendung auf dem entfernten Netzwerkgerät angesprochen und am Quell-Port auf die Antwort gewartet. Die Ports 0 bis 1023 sind dabei festen Anwendungen zugeordnet. Die restlichen 16 Bit großen Ports können beliebig von anderen Programmen genutzt werden.

Das wichtigste Protokoll der Transportschicht ist das **Transmission Control Protocol (TCP)** [51], das eine zuverlässige, verbindungsorientierte Kommunikation zwischen zwei Netzwerkteilnehmern ermöglicht. TCP übernimmt während der Datenübertragung auch Aufgaben der Datenflusssteuerung und ergreift Maßnahmen bei einem Datenverlust. Dazu teilt TCP die Daten vor dem Versenden in Datenblöcke auf, versieht sie mit einem TCP-Header und übergibt sie der Internetschicht. Der TCP-Header enthält unter anderem eine Sequenznummer, die am Anfang einer Verbindung zwischen den Kommunikationspartnern ausgehandelt und während der Kommunikation fortlaufend erhöht wird. Beim Empfänger werden die ankommenden Datenblöcke sortiert und wieder zu einem Datenstrom zusammengefügt, bevor

sie anhand der Portnummer an die zugehörige Anwendung übergeben werden. Auf diese Weise bleibt die Reihenfolge der Datenblöcke erhalten. Weiterhin können verloren gegangene Datenblöcke anhand der Sequenznummer erkannt und neu angefordert werden. Ein Datenverlust ist bei TCP somit ausgeschlossen.

Neben dem verbindungsorientierten TCP gehört aber auch das verbindungslose Datagramm-Protokoll **User Datagram Protocol (UDP)** [52] in die Transportschicht des TCP/IP-Modells. Als ein deutlich einfacheres Protokoll arbeitet es ohne eine Nummerierung der Datenpakete und besitzt somit keine Methoden um die Reihenfolge der übertragenen Pakete bzw. überhaupt deren Empfang sicherzustellen. UDP überprüft lediglich die Korrektheit der empfangenen Datenpakete und leitet fehlerfreie Pakete durch Nutzung der Port-Struktur an die richtige Anwendung weiter. Die Aufgabe einer zuverlässigen Datenübertragung mit erneuter Übertragung fehlerhafter Pakete sowie die Flusssteuerung der Kommunikationsverbindung wird der jeweiligen Anwendung selbst überlassen. Anwendungen, die UDP zur Datenübertragung nutzen, sind häufig im Bereich der Übertragung von Audio- und Video-Daten anzutreffen, da kleine Übertragungsfehler hier kaum merkbar sind und zeitaufwändige Neuübertragungen eher störend wirken können.

Network Address Translation

Der Internet-Boom in der Mitte der neunziger Jahre führte zu einer rasant steigenden Nachfrage an Internet-Zugängen und Webservern. Mit jedem neuen Teilnehmer dieses IP-basierten Netzes stieg die Menge der benötigten IP-Adressen an. Da der 32 Bit große Adressraum der verwendeten IPv4-Adressen jedoch auf 2^{32} Adressen begrenzt ist, erwies er sich schnell als zu klein um jedem Computer, der einen Zugang zum Internet besitzt, eine IP-Adresse bereitzustellen. Um das Problem der knappen IP-Adressen zu umgehen, wurde im Jahre 1994 eine Technik namens *Network Address Translation* (NAT) entwickelt und im RFC 1631 [53] (jetzt [54]) standardisiert, die eine strikte Trennung der IP-Adressbereiche zwischen dem öffentlichen Internet und dem privaten Netzwerk einführte. Für die Datenvermittlung zwischen beiden Netzwerken werden sogenannte NAT-Router eingesetzt, die ausgestattet mit zwei Netzwerkschnittstellen sowohl Teilnehmer des öffentlichen Internets sind und hierbei eine öffentliche IP-Adresse nutzen, als auch Teilnehmer eines lokalen, privaten Netzes sind und hierfür eine private IP-Adresse besitzen. Diese Technik hat den Vorteil, dass die Teilnehmer des privaten Netzwerks mit einem beliebigen IP-Adressbereich konfiguriert werden können und für die Kommunikation mit einem Server im Internet nur eine öffentliche IP-Adresse, nämlich die des NAT-Gerätes, stellvertretend für das gesamte Netzwerk benötigt wird. Um Konflikte zwischen öffentlichen und

privaten IP-Adressen zu vermeiden, wurden durch den RFC 1918 [55] für die Verwendung privater IP-Adressen die speziellen Adressblöcke 10.0.0.0-10.255.255.255, 172.16.0.0-172.31.255.255 und 192.168.0.0-192.168.255.255 reserviert, die im öffentlichen Internet weder verwendet noch geroutet werden.

Während der in Abbildung 2.4 dargestellten Kommunikation zwischen Teilnehmern des privaten Netzes und dem öffentlichen Internet, stellt sich der NAT-Router gegenüber dem Internet als Absender aller Anfragen dar. Dazu muss er die ausgehenden Datenpakete so modifizieren, als ob sie von ihm selbst geschickt wurden. In der ausgehenden Richtung der Datenpakete, bei der Pakete vom lokalen Netzwerk in das Internet übertragen werden sollen, wird die sogenannte *Source Network Address Translation* (SNAT) angewendet, die in allen ausgehenden Paketen die Quell-IP-Adresse durch die öffentliche IP-Adresse des Routers ersetzt. In der umgekehrten Richtung sorgt die *Destination Network Address Translation* (DNAT) dafür, dass vom Internet eingehende Pakete wieder an das richtige Ziel im privaten Netz weitergeleitet werden. Hierbei ist die Ziel-IP-Adresse, in der die öffentliche IP-Adresse des Routers eingetragen ist, durch die private IP-Adresse des lokalen Teilnehmers zu ersetzen.

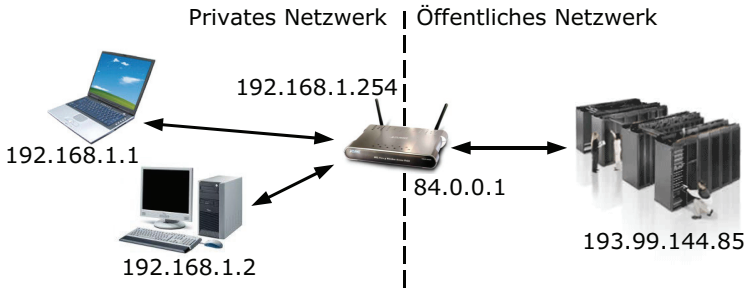

Abbildung 2.4 Szenario einer Network Address Translation

Port Address Translation

Mit den Namen *Masquerading*, *Port Address Translation* (PAT), *Network Address Port Translation* (NAPT) und auch *1-to-n-NAT* wird eine spezielle Form von NAT verstanden, die alle Adressen eines privaten Netzwerks auf eine einzelne öffentliche IP-Adresse abbildet und dabei nicht nur die IP-Adresse sondern auch die Port-Nummer austauscht. Angenommen, der in Abbildung 2.4 dargestellte Router besitzt die öffentliche IP-Adresse 84.0.0.1, die beiden Teilnehmer des privaten Netzes die Adressen 192.168.1.1 sowie 192.168.1.2 und der Server im Internet die Adresse 193.99.144.85. Für die Kommunikation zwischen dem Laptop mit privater Adresse und dem öffentlich erreichbaren Server besteht die Aufgabe des Routers nun darin, in jedem

ausgehenden Paket die Quell-IP-Adresse 192.168.1.1 durch seine eigene öffentliche IP-Adresse 84.0.0.1 zu ersetzen. Weiterhin muss der Quell-Port (3000) des Paketes, wie in Abbildung 2.5 dargestellt, durch einen neuen freien Port (32768) ersetzt werden, der es später erlaubt, eingehende Pakete eindeutig dem zugehörigen lokalen Gerät zuzuordnen.

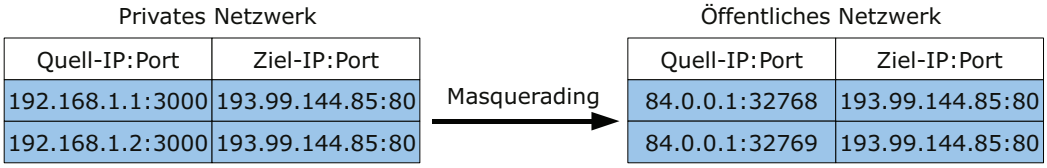

Abbildung 2.5 Port Address Translation für ausgehende Pakete

Mit Hilfe einer NAT-Tabelle merkt sich der Router die Zuordnung der privaten IP-Adresse des Rechners, der das Paket verschickt hat, die Quell-Port-Nummer und die vergebene öffentliche Port-Nummer. Nachdem auch der PC eine Verbindung zum Server aufgebaut hat, besitzt die NAT-Tabelle die in Abbildung 2.6 dargestellten Einträge.

Private IP	Öffentliche IP	Privater Port	Öffentlicher Port
192.168.1.1	84.0.0.1	3000	32768
192.168.1.2	84.0.0.1	3000	32769

Abbildung 2.6 Beispiel einer NAT-Tabelle für eine Port Address Translation

Anschließend kann bei eingehenden Paketen anhand der Ziel-Port-Nummer der dazugehörige Eintrag aus der NAT-Tabelle ermittelt werden, der die IP-Adresse des lokalen Rechners sowie den ursprünglichen Port enthält. Mit diesen Informationen wird dann, wie in Abbildung 2.7 dargestellt, die Zieladresse (84.0.0.1) und der Ziel-Port (32768) durch die IP-Adresse (192.168.1.1) und den originalen Port (3000) ersetzt und an den Teilnehmer im privaten Netzwerk weitergeleitet.

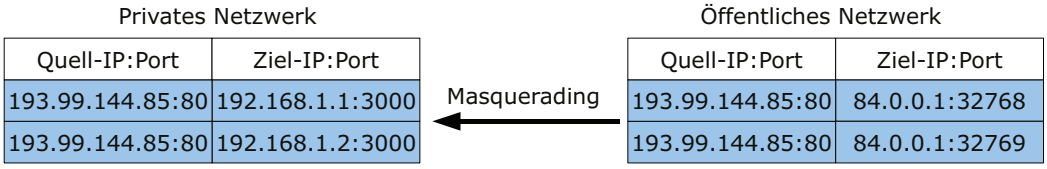

Abbildung 2.7 Port Address Translation für eingehende Pakete

Masquerading hat den Nachteil, dass direkte Verbindungen vom Internet zu Servern im privaten Netzwerk nur durch ein Port Forwarding möglich sind. Dabei werden Pa-

kete, die an frei wählbare Ports des Routers adressiert sind, durch statische Einträge im Router an einen bestimmten Rechner innerhalb des privaten Netzes weitergeleitet. Zum Beispiel können so Pakete, die an den Port 80 adressiert sind, an einen Webserver im privaten Netzwerk vermittelt werden. Die Erreichbarkeit zweier unterschiedlicher Webserver im privaten Netz, die den gleichen Port nutzen, ist hierbei jedoch vom Internet aus nicht möglich.

Basic-NAT

Eine andere Form der Network Address Translation ist das *Basic-NAT*, bei dem jede interne IP-Adresse durch eine externe IP-Adresse ersetzt wird. Bei dieser 1:1-Übersetzung wird in den ausgehenden Paketen die private Quell-IP-Adresse wie in Abbildung 2.8 durch eine noch nicht benutzte öffentliche IP-Adresse ersetzt.

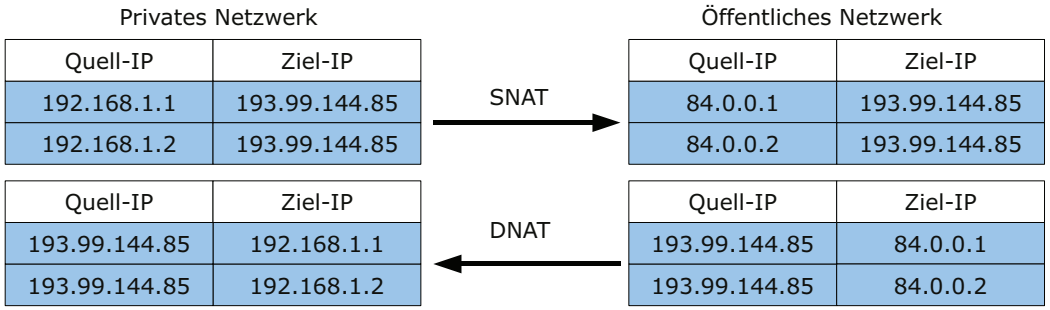

Abbildung 2.8 Funktionsweise eines Basic-NAT-Routers

Daraus ergibt sich eine NAT-Tabelle mit den in Abbildung 2.9 dargestellten Einträgen. Mit Hilfe dieser Einträge wird für Pakete, die aus dem öffentlichen Netzwerk am NAT-Router ankommen, je nach Destination-IP-Adresse des Pakets die zugehörige private Adresse ermittelt, im Paket ersetzt und das Paket an den lokalen Rechner weitergeleitet. Basic-NAT bietet gegenüber Masquerading den Vorteil, dass unter Umständen auch mehrere Server unter derselben Portnummer vom Internet aus erreichbar sind und die interne Struktur des privaten Netzwerks gleichzeitig nach außen verborgen bleibt.

Private IP	Öffentliche IP	Privater Port	Öffentlicher Port
192.168.1.1	84.0.0.1	-	-
192.168.1.2	84.0.0.2	-	-

Abbildung 2.9 Beispiel einer NAT-Tabelle für Basic-NAT

Das größte Problem an NAT ist, dass eine saubere Trennung der Netzwerkschichten nicht eingehalten wird. Es fällt besonders bei der Entwicklung neuer Protokolle der Anwendungsschicht des TCP/IP-Referenzmodells auf, wenn Protokolle, wie z. B. das *Session Initiation Protocol* (SIP), das vorwiegend für die *Voice over IP* (VoIP)-Telefonie eingesetzt wird, nur dann durch einen NAT-Router hindurch funktionieren, wenn er diese Protokolle explizit unterstützt. Die Entwicklung neuer Netzwerk-Protokolle wird dadurch deutlich erschwert. Mit der zügigen Einführung von IPv6 und dem damit verbundenen erweiterten Adressbereich ist es aber kein Problem mehr jedem Computer eine eindeutige IP-Adresse zuzuweisen, was NAT in IPv6-Netzen faktisch unnötig macht und wieder eine saubere Trennung der Netzwerkschichten ermöglicht.

2.2.3 Protokolle zur Konfiguration IP-basierter-Netze

Aufbauend auf den UDP- und TCP-Verbindungen der Transportschicht, übernimmt die Anwendungsschicht als oberste Schicht des TCP/IP-Modells die Aufgaben des Session, Presentation und Application Layers des OSI-Referenzmodells. Sie umfasst somit alle Protokolle, die für die Konfiguration eines Netzwerks benötigt werden oder die Netzwerkinfrastruktur für den Austausch anwendungsspezifischer Daten nutzen. In diese Schicht lassen sich anwendungsorientierte Protokolle wie z. B. HTTP, HTTPS, FTP, SSH und SOAP; aber auch die Protokolle DHCP, DNS und ARP einordnen, die mit der Vergabe und Auflösung von IP-Adressen, der Netzwerkkonfiguration dienen und somit als Hilfsprotokolle in IP-basierten Netzen eingesetzt werden. Da auf einige dieser Protokolle im Konzeptkapitel der vorliegenden Arbeit zurückgegriffen wird, beschreiben die nun folgenden Abschnitte ausgewählte Protokolle detaillierter.

Bei der Einbindung eines neuen Computers in ein bestehendes Netzwerk steht der Computer vor dem Problem, dass er zwar eine MAC-Adresse besitzt, die seiner Netzwerkschnittstelle fest zugeordnet ist, ihm aber für eine IP-basierte Kommunikation eine IP-Adresse fehlt, die im lokalen Netzwerk eindeutig ist. Für eine möglichst einfache Konfiguration neuer Netzwerkteilnehmer wurden daher unterschiedliche Verfahren entwickelt, die eine automatische Zuweisung einer IP-Adresse sowie weiterer Netzwerkparameter erlauben. Diese Verfahren basieren auf dem Client-Server-Konzept, bei dem Clients eine Anfrage an einen Server stellen, der für die Vergabe einer eindeutigen IP-Adresse zuständig ist.

Das **Reverse Address Resolution Protocol (RARP)** [56] ist eines der ersten Verfahren dieser Art. Es ermöglicht eine statische Zuordnung von MAC- und

IP-Adressen, die durch eine manuelle Konfiguration in einer Tabelle beim RARP-Server hinterlegt ist. Um eine IP-Adresse zu erhalten, sendet der RARP-Client einen RARP-Request-Broadcast, der die eigene MAC-Adresse enthält, an die am Netzwerk angeschlossenen Rechner. Der RARP-Server sucht aus der Tabelle mit den IP-MAC-Zuordnungen die dort hinterlegte IP-Adresse heraus und sendet sie in einem RARP-Reply an den Client zurück. Neben der statischen Adresszuordnung hat RARP den Nachteil, dass die Übertragung weiterer Konfigurationsparameter an den Client nicht vorgesehen ist und das Verfahren durch Verwendung des MAC-Broadcasts auf den Netzwerkbereich bis zum nächsten Router eingeschränkt ist.

Das auf RARP aufbauende **Bootstrap Protocol (BOOTP)** [57] vermeidet diese Nachteile durch den Einsatz eines Forwarding Agents auf dem Router, der BOOTP-Broadcasts auch in andere Netzwerke weiterleitet. Auf diese Weise entfällt die Notwendigkeit eines RARP-Servers in jedem Subnetz. Weiterhin können mit BOOTP zusätzliche Parameter an den Client übergeben werden, die eine vollständige Konfiguration seiner Netzwerkschnittstelle ermöglichen.

Auch das **Dynamic Host Configuration Protocol (DHCP)** [58] arbeitet als Erweiterung des Bootstrap-Protokolls nach einem Client-Server-Prinzip, um neu hinzukommenden Geräten eine Netzwerkkonfiguration zuzuweisen. DHCP wurde jedoch für eine automatische Einbindung neuer Geräte in große Netzwerke mit häufig wechselnder Topologie entwickelt. Es bietet den Vorteil, dass die Zuordnungen von IP- und MAC-Adressen nicht mehr statisch durch den Administrator des Netzwerks festgelegt werden müssen, sondern IP-Adressen dynamisch aus einem bestimmten Adressbereich vergeben werden können. Für die Realisierung dieses Ansatzes wird beim DHCP-Server eine Konfigurationsdatei hinterlegt, die Informationen über die zu vergebenen IP-Adressen und zusätzliche Netzwerk-Parameter (wie z. B. die Subnetzmaske, den DNS-Server oder das Default Gateway) enthält. DHCP setzt für die Client-Server-Kommunikation das UDP-Protokoll ein. Der DHCP-Server wartet auf dem UDP-Port 67 auf Anfragen von Clients. Damit Clients ihre Netzwerkkonfiguration bekommen, schicken sie eine DHCPDISCOVER-Nachricht, die ihre MAC-Adresse enthält, als IP-Broadcast an das lokale Netzwerk. Da der Absender noch keine IP-Adresse besitzt, ist die Quell-IP-Adresse im UDP-Paket dieser Nachricht auf 0.0.0.0 gesetzt. Die Ziel-IP-Adresse ist auf die Broadcast-Adresse 255.255.255.255 gesetzt und führt dazu, dass alle Teilnehmer des lokalen Netzes diese Nachricht empfangen. DHCP-Server empfangen das an Port 67 adressierte Paket, machen einen Vorschlag für eine IP-Adresse und antworten dem Client mit einem DHCPOFFER, das ebenfalls die Broadcast-Adresse 255.255.255.255 als Ziel-IP-Adresse enthält. Sind mehrere DHCP-Server in einem lokalen Netzwerk vorhanden, wählt der Client ein auf Port 68 empfangenes DHCPOFFER aus und kontaktiert den ausgewählten Server mit einem DHCPREQUEST. Dieses Paket wird ebenfalls als Broadcast gesendet, damit die an-

deren DHCP-Server die Auswahl des Clients mitbekommen und es als Absage ihrer Angebote werten können. Anschließend bestätigt der ausgewählte DHCP-Server die IP-Adresse mit einer DHCPACK-Nachricht und übersendet die weiteren relevanten Daten an den Client.

Beim Einsatz von DHCP zur automatischen Einbindung neuer Geräte können unterschiedliche Betriebsmodi für die Vergabe der IP-Adressen verwendet werden. Die *manuelle Zuordnung* basiert auf einer festen Zuordnung von IP-Adressen zu bestimmten MAC-Adressen. Diese Adressvergabe ist für Server-Anwendungen wichtig, die für eine unbestimmte Zeit unter einer festen IP-Adresse erreichbar sein sollen. Sie dient aber auch einer Art Teilnahmekontrolle, die anhand der MAC-Adresse festlegt, welche Clients eine IP-Adresse bekommen und das Netzwerk nutzen dürfen. Die automatische und die dynamische Zuordnung eliminieren die Notwendigkeit der statischen Konfiguration von IP-Adressen. Bei der *automatischen Zuordnung* wird dazu ein Bereich von IP-Adressen definiert und eine freie IP-Adresse automatisch für eine MAC-Adresse vergeben. Die Zuordnung von IP- und MAC-Adresse wird dabei gespeichert und führt dazu, dass eine IP-Adresse nicht neu vergeben wird, auch wenn ein Netzwerkteilnehmer das Netzwerk verlässt. Bei der *dynamischen Zuordnung* wird eine IP-Adresse für eine bestimmte Zeit an einen Client vermietet. Neigt sich diese sogenannte Lease Time dem Ende, muss sich der Client erneut beim DHCP-Server melden und eine Verlängerung durch einen erneuten DHCP-Request beantragen. Tut er dies nicht, wird die IP-Adresse frei und kann an einen anderen Client vergeben werden.

Bei der Kommunikation in IP-basierten Netzen wird davon ausgegangen, dass alle Netzwerkteilnehmer eindeutig durch ihre IP-Adresse adressiert werden können. Für den Nutzer eines Computers ist es jedoch schwer, die IP-Adressen aller Kommunikationspartner zu merken. Es ist für ihn leichter, jedem Computer einen Namen zuzuordnen, der einfacher zu lesen und einzuprägen ist. In TCP/IP-Netzwerken wird daher jedem Teilnehmer ein Name zugeordnet, der eine eindeutige Bezeichnung innerhalb eines lokalen Netzwerks ermöglicht. Dieses Netzwerk wird mit einem Domainnamen benannt, der aus verschiedenen Subdomainen bestehen kann und eine eindeutige Bezeichnung im weltweiten Internet zulässt. Aus der Kombination beider Namen ergibt sich ein weltweit eindeutiger Hostname [59], wie z. B. weser.informatik.uni-rostock.de, der den Namen (weser) und die zugehörige Domain (informatik.uni-rostock.de) des Computers enthält. Durch eine Namensauflösung kann dieser Hostname in die IP-Adresse 139.30.7.100 aufgelöst und für die IP-basierte Kommunikation verwendet werden. Bei der Namensauflösung gibt es unterschiedliche Ansätze. So können die Namen und die zugehörigen IP-Adressen in einer Hosts-Datei auf dem lokalen Computer hinterlegt oder aber auch ein spezieller **Domain Name System (DNS)**-Dienst [60] im lokalen Netzwerk genutzt werden, bei dem die verwendeten Namen und dazu-

gehörigen IP-Adressen statisch bzw. dynamisch einzutragen sind. Bei der Nutzung von DNS werden sogenannte Nameserver eingesetzt, die jeweils für eine Subdomaine zuständig und hierarchisch in Baumform organisiert sind. Möchte ein Netzwerkteilnehmer also die IP-Adresse des Hostnamens weser.informatik.uni-rostock.de wissen, schaut er zunächst in seiner Hosts-Datei nach, ob der Name und die IP-Adresse dort hinterlegt sind. Falls nicht, fragt er beim lokalen DNS-Server nach, dessen Adresse er zuvor vom DHCP-Server während seiner Konfiguration bekommen hat. Liegt der Namensraum des angefragten Hostnamens außerhalb der eigenen Domäne, leitet der Nameserver die Anfrage solange an einen hierarchisch höheren Nameserver weiter, bis er entweder eine positive oder eine negative Antwort eines Root-Nameservers erhält. Eine positive Antwort ermöglicht ihm anschließend, die IP-Adresse des Kommunikationspartners wie gewohnt für beliebige Protokolle der Anwendungsschicht zu nutzen. Da einem Hostnamen auch mehrere IP-Adressen zugeordnet werden können, ist mit DNS sogar eine Lastverteilung auf z. B. mehrere HTTP-Server möglich.

Eine IP-basierte Ende-zu-Ende-Kommunikation, die durch die Internetschicht ermöglicht wird, basiert auf Punkt-zu-Punkt-Verbindungen, die durch die Netzzugangsschicht bereitgestellt werden. Daher reicht es für eine IP-basierte Kommunikation nicht aus, die IP-Adresse des Kommunikationspartners zu kennen. Vielmehr muss auch entweder die MAC-Adresse des lokal erreichbaren Kommunikationspartners oder die MAC-Adresse des nächsten Zwischenknotens auf dem Weg zum entfernten Kommunikationspartner im anderen Subnetz bekannt sein. Zur Bestimmung dieser MAC-Adresse, die im Ethernet-Header jedes Datenpakets als Zieladresse eingetragen werden muss, ist das **Address Resolution Protocol (ARP)** [61] zuständig. Auch ARP arbeitet nach dem Client-Server-Prinzip und nutzt MAC-Broadcasts zum Informationsaustausch. Möchte der Client die MAC-Adresse zu einer IP-Adresse in Erfahrung bringen, sendet er einen ARP-Request als MAC-Broadcast an das lokale Netzwerk, der die IP-Adresse des gesuchten Computers enthält. Im Ethernet-Header des Requests ist dabei seine eigene MAC-Adresse als Quelladresse und 255.255.255.255 als Zieladresse eingetragen. Befindet sich der gesuchte Computer im selben Subnetz, kann er seine MAC-Adresse durch einen ARP-Reply an den anfragenden Client zurücksenden. Kann jedoch anhand der IP-Adresse erkannt werden, dass sich der gesuchte Computer in einem anderen Subnetz befindet, wird stellvertretend die MAC-Adresse des Routers verwendet, über den der gesuchte Computer erreichbar ist. Anschließend trägt der Anfragende die Kombination aus IP- und MAC-Adresse in seine lokale ARP-Tabelle ein. Damit stehen sowohl die IP-Adresse des Empfängers als auch die MAC-Adresse des nächsten Hops für die Kommunikation höherer Protokollschichten bereit.

2.2.4 Service-basierte Kommunikation

Service-orientierte Architektur

Mit der Entwicklung zunehmend verteilter und komplexerer Kommunikationssysteme stiegen die Ansprüche an die Anwendungsprotokolle und vor allem an die Art der Kommunikation. Dies führte zu dem Wunsch, von den konkreten Funktionen der Kommunikationspartner zu abstrahieren und sie in Form sogenannter Dienste (engl. Services) zu nutzen. Dadurch lassen sich Anwendungen funktional zerlegen und prozessorientiert betrachten. Dienste sind in sich geschlossen und können eigenständig genutzt werden. Sie verfügen über eine öffentliche Schnittstelle, die die Komplexität des Dienstes sowie Realisierungsdetails verbirgt. Weiterhin beschreibt sie, welche Eingaben für die Dienstnutzung erforderlich sind und welcher Art das Ergebnis ist. Da Dienste erst bei der Ausführung dynamisch lokalisiert und eingebunden werden, steht ihrer Austauschbarkeit mit einem Dienst, der über die gleiche abstrakte Schnittstelle verfügt, nichts im Wege. Mit der Abstraktion von Funktionalitäten in Form von Diensten entwickelte sich das Konzept einer *Service-orientierten Architektur* (SOA), das sich mit dem Anbieten, Suchen und Nutzen von Diensten über ein Netzwerk befasst. An SOA wurden in den letzten Jahren große Erwartungen gestellt, die nach einer gewissen Praxisdauer zu einer neutralen und objektiven Neubewertung führten. Derzeit stellt SOA einen vielversprechenden Weg dar, der durch eine Kapselung von Funktionalitäten in Dienste und die Trennung von Schnittstelle und Implementierung eine hohe Wiederverwendbarkeit, eine vereinfachte Wartung und sogar die Orchestrierung zu komplexeren Diensten einer höheren Abstraktionsebene erlaubt. Gleichzeitig bietet sie eine spontane, flexible und transparente Einbindung neuer Geräte und Anwendungen zum Zeitpunkt des Bedarfs.

Die Merkmale einer SOA sind in Abbildung 2.10 in Form eines SOA-Tempels dargestellt. Als Basis dieser Architektur dient die Verwendung offener *Kommunikationsstandards*, die zu einer problemlosen Verständigung der Kommunikationspartner führt. Darauf aufbauend kommt der *Sicherheit* der zu übertragenden XML-basierten Daten eine tragende Bedeutung zu, die durch etablierte Standards wie z. B. XML-Encryption erfüllt werden kann. Ein weiteres Merkmal einer SOA ist die *Einfachheit*, die dadurch erreicht werden kann, dass die Schnittstelle und die Implementierung der Dienste getrennt und Dienste in verschiedenen Umgebungen mehrfach ohne hohen Aufwand eingesetzt werden. Weitere Merkmale sind die *Verteiltheit* und die *Prozessorientiertheit*, die durch die Bereitstellung von Diensten durch verschiedene Server sowie die zum Großteil autark ablaufende Kommunikation zwischen ihnen erreicht wird. Dabei können grobe Prozessabläufe im Voraus modelliert und durch Ereignisse, wie z. B. die Interaktion eines Menschen, gesteuert werden. Als optionaler Bestand-

teil dient der *Verzeichnisdienst* der Registrierung verfügbarer Dienste und ermöglicht anderen Anwendungen, dynamisch nach Diensten zu suchen. Durch diese *Lose Kopplung* ist es möglich, geeignete Dienste erst zur Laufzeit der Anwendung zu suchen und gefundene Dienste dynamisch einzubinden.

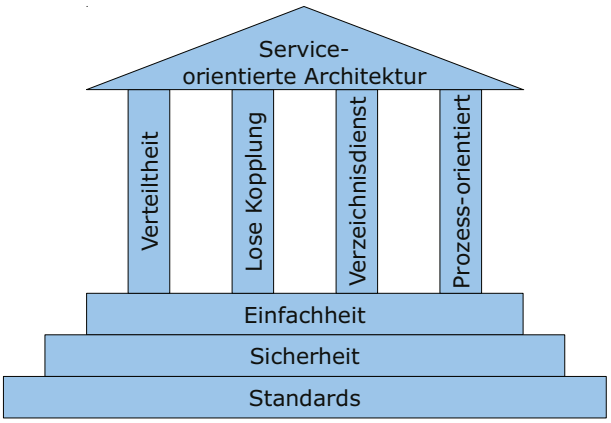

Abbildung 2.10 Merkmale einer Service-orientierten Architektur

Die Komponenten einer SOA sind in drei Rollen eingeteilt, die, wie in Abbildung 2.11 dargestellt, zusammenarbeiten.

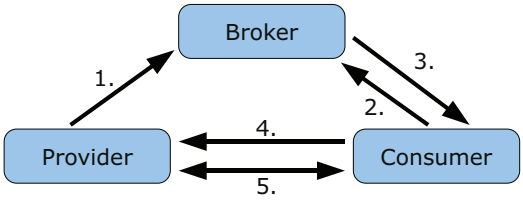

Abbildung 2.11 Rollen in einer Service-orientierten Architektur

Der *Provider* (Dienstanbieter) stellt seine Funktionalität als Dienst bereit und registriert ihn mit seiner öffentlichen Schnittstelle bei einem Dienstverzeichnis (1.). Dieses Verzeichnis übernimmt die Rolle des *Brokers* (Vermittlers) zwischen dem Dienstanbieter und dem Dienstnutzer und ermöglicht die gezielte Suche nach Diensten. Zur Nutzung eines Dienstes stellt der *Consumer* (Dienstnutzer) eine Anfrage an den Verzeichnisdienst (2.) und bekommt einen Verweis zum passenden Dienstanbieter zurück (3.), über den er Details des Dienstes abfragen kann (4.). Anschließend kann der Dienst direkt genutzt werden (5.). Für die Funktion einer SOA, die durchaus in unterschiedlichen Programmiersprachen und auf verschiedenen Plattformen realisiert sein kann, sind Dienstanbieter und Dienstnutzer obligatorisch; der Vermittler ist je nach eingesetzter SOA-Technologie optional und vorwiegend bei einer hohen Anzahl an Dienstanbietern nützlich.

Web Services

Eine konkrete und bereits bewährte Implementierung des SOA-Paradigmas stellen die sogenannten *Web Services* (WS) dar, deren Komponenten, wie im vorherigen Abschnitt erläutert, die Rollen des Consumers, Providers und Brokers einnehmen. Mit der WS-Spezifikation [62] ist die Kommunikation zwischen Consumern und Providern sowie die Funktionalität des Brokers standardisiert. Das Design der Dienste wird durch die Spezifikation nicht vorgegeben. Bei einem Web Service handelt es sich um eine in sich gekapselte Anwendung, die eine genau spezifizierte Aufgabe erfüllt. Web Services können nur durch die Verwendung ihrer Schnittstelle genutzt werden. Sie verfügen über kein graphisches Interface, da sie nur für die Kommunikation zwischen Anwendungen gedacht sind (Maschine-Maschine-Kommunikation) und nicht zur Interaktion mit dem Menschen. Vielmehr ist die Anwendung, in der sie eingesetzt werden, dafür zuständig. Web Services sind zu jedem Zeitpunkt und von jedem Ort abrufbar und können zu größeren Diensten kombiniert werden. Die Grundlage der WS-Spezifikation bilden die drei XML-basierten Standards SOAP, WSDL und WSIL bzw. UDDI. Dabei dient WSDL der Schnittstellenbeschreibung, die aufzeigt, wie und mit welchen Daten der Dienst angesprochen und in welchem Format die Antwort erwartet werden kann. Die lose Kopplung des Dienstes wird durch einen SOAP-basierten Nachrichtenaustausch erreicht, bei dem weder Provider noch Consumer über Implementierungsdetails des jeweils anderen verfügen müssen. Auch für die Kommunikation mit dem WSIL-Dienstverzeichnis wird SOAP als Anwendungsprotokoll genutzt.

Als leichtgewichtiges Standardprotokoll zur Übermittlung von XML-Nachrichten stellt **SOAP** [63] Regeln für den Aufbau der Nachrichten auf und gibt somit vor, wie Daten innerhalb einer Nachricht abzubilden und zu interpretieren sind. SOAP-Nachrichten entsprechen einer XML-Datei, die aus den Teilen *SOAP Envelope*, *SOAP Header* und *SOAP Body* besteht. Der SOAP Envelope ist das Wurzelelement der XML-Nachricht. Er enthält die Angabe der verwendeten Version der SOAP-Spezifikation und legt mit dem Namensraum die Schemadefinition des Dokuments fest. Der Envelope schließt weiterhin einen optionalen SOAP Header und einen SOAP Body ein. Der SOAP Header kann Authentisierungs- und Transaktionsinformationen enthalten. Im SOAP Body sind die zu übertragenden Information enthalten, deren Struktur von den Kommunikationspartnern im Voraus durch die öffentliche Schnittstelle des Dienstes festgelegt wurde. Hierbei sind auch Funktionsaufrufe und entsprechende Rückantworten möglich, die durch die Schnittstelle des Dienstes definiert werden. Aufbauend auf diesem Nachrichtenformat enthält die SOAP-Spezifikation drei Arten an Nachrichten, die entweder für eine Anfrage, eine Antwort oder eine Fehlermeldung eingesetzt werden können. Eine Anfrage enthält den Funktionsaufruf

und dessen eventuell benötigte Parameter. Bei erfolgreichem Aufruf gibt die Antwort-Nachricht das Ergebnis des Funktionsaufrufs zurück. Tritt ein Fehler im Funktionsaufruf oder bei der Funktionsausführung auf, beinhaltet die Fehlernachricht eine ausführliche Fehlerbeschreibung. Für die Übermittlung der SOAP-Nachrichten wird in der Regel HTTP oder HTTPS eingesetzt, obwohl auch die Nutzung anderer Protokolle wie FTP oder SMTP möglich ist.

Die vom W3C festgelegte **Web Services Description Language (WSDL)** [64] dient der Schnittstellenbeschreibung der Dienste in Form eines XML-basierten Dokuments, das beim Provider bereitgestellt wird. Durch die WSDL-Beschreibung kann ein Consumer nähere Informationen zum Dienst erhalten. Sie gibt an, wie und mit welchen Daten ein Dienst angesprochen und in welchem Format die Antwort erwartet werden kann. Dabei werden verschiedene Hauptelemente eingesetzt, die den Dienst näher definieren. So legt z. B. *binding* das konkrete Protokoll und Datenformat für die Interaktion mit dem Service fest. Ein sogenannter *endpoint* definiert die Adresse unter der der Dienst verfügbar ist. Die Sequenz von Nachrichten, die ein Service sendet oder empfängt werden im Element *operations* gruppiert und mehrere von ihnen durch ein *interface* zusammengefasst. WSDL-Dokumente können durch zusätzliche Elemente wie z. B. *include*, *import* oder *feature* erweitert werden.

Für Web Services gibt es unterschiedliche Verzeichnisdienste, die als Broker eingesetzt werden können. Der wohl bekannteste und auch komplexeste ist *Universal Description, Discovery and Integration* (UDDI) [65]. Dieser plattformunabhängige, zentrale Verzeichnisdienst eignet sich vor allem für internationale Unternehmen, die ihre Services mit zusätzlichen Informationen in mehreren Verzeichnissen geordnet ablegen möchten. Aufgrund seiner Komplexität wurde UDDI jedoch nur schwer von Entwicklern angenommen und konnte sich deshalb nicht durchsetzen. Neben UDDI ist die **Web Services Inspection Language (WSIL)** [66] ein wichtiger Verzeichnisdienst für Service-orientierte Architekturen. WS-Inspection nutzt ein einfaches Konzept zur Dienstsuche und arbeitet auf Basis kleiner, zentraler Verzeichnisse, die nicht nur den Ort und die bereitgestellten Funktionen eines Web Services spezifizieren, sondern auch wie diese genutzt werden können. Dabei werden der Dienstname, eine kurze Zusammenfassung der Funktionen des Dienstes und die Adresse des Dienstanbieters, der die WSDL-Beschreibung des Dienstes veröffentlicht, in ein XML-Dokument eingetragen, das der Consumer unter Verwendung des HTTP-Protokolls beim Broker abruft und sich daraus den Verweis auf einen beliebigen Dienst heraussucht. Dies setzt voraus, dass den Consumern die IP-Adresse des Brokers bekannt ist. Da diese Architektur keine Möglichkeit bietet, die Broker-Adresse zu ermitteln, ist eine spontane Vernetzung nur mit dieser Einschränkung möglich.

2.3 Adressierung und Kommunikation in nicht-IP-basierten Netzen

Für die zukünftige Entwicklung der mobilen Kommunikation haben gerade die Technologien eine besondere Bedeutung, die eine spontane und direkte Vernetzung der Geräte ermöglichen, die uns in unserem ganz persönlichen Umfeld umgeben. Die verbreitetsten WPAN-Technologien in diesem Bereich sind Bluetooth und ZigBee, die aufgrund ihrer bereits vorgestellten Netzwerktopologie keine IP-Adressen benötigen. Sie stellen zwar nur geringe Datenraten bereit, aber durch den Einsatz geringer Sendeleistungen sowie spezieller Adressierungs- und Kommunikationsmechanismen arbeiten sie ernergiesparsamer als z. B. WLAN und verlängern die Akkulaufzeit der mobilen Geräte deutlich.

2.3.1 Adressbasierte Kommunikation in Bluetooth

Der Bluetooth Protokollstack

Ähnlich dem TCP/IP-Referenzmodell, das in IP-basierten Netzwerken Mechanismen für die Adressierung der Netzwerkteilnehmer und die Kommunikation zwischen ihnen in verschiedene Protokollschichten aufteilt, gibt es auch in Bluetooth-Netzen einen Protokollstack, der unterschiedliche Aufgaben in verschiedene Schichten unterteilt. Der Bluetooth Protokollstack ist in Abbildung 2.12 [31] dargestellt. Seine Komponenten lassen sich grob in den *Host*, den *Controller* und das *Host Controller Interface* (HCI) einteilen. Der Controller stellt die Hardware einer Bluetooth-Schnittstelle in einem Gerät dar. Das HCI entspricht dem Bluetooth-Treiber im Betriebssystem und ermöglicht den höheren Softwarekomponenten des Hosts über einheitliche Schnittstellen auf den Controller zuzugreifen.

Da der Bluetooth-Stack vor allem für die Punkt-zu-Punkt-Kommunikation entworfen wurde, lässt er sich nicht direkt auf das TCP/IP-Referenzmodell abbilden [35]. Es lassen sich aber starke Ähnlichkeiten erkennen, wobei sich das *Baseband*, der *Link Manager* und der *Physical Layer* (PHY) des Controllers in die Netzzugangsschicht des TCP/IP-Modells einordnen lassen. Das *L2CAP* hat für Bluetooth eine ähnliche Bedeutung wie die Internetschicht für das TCP/IP-Modell. RFCOMM ermöglicht durch die Nutzung von sogenannten Kanälen ein Multiplexing, wie auch die zuvor beschriebenen TCP- und UDP-Verbindungen der Transportschicht in IP-basierten Netzen. Die übrigen Protokolle des Bluetooth-Stacks wie z. B. *OBEX* und *SDP* sind der Anwendungsschicht zuordenbar.

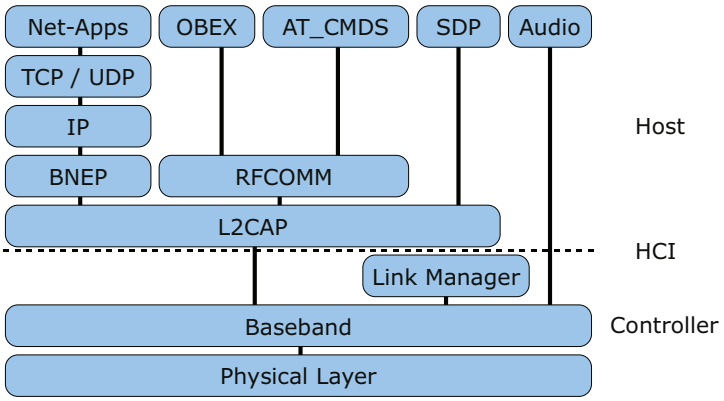

Abbildung 2.12 Aufbau des Bluetooth-Stacks [31]

Der **Physical Layer**, als unterste Schicht des Bluetooth-Stacks, legt die Frequenz und die Sendeleistung der Funkschnittstelle fest. Auch das von Bluetooth eingesetzte Frequency Hopping unter Nutzung von 79 Kanälen im 2,4 GHz-Band, das zu einer Robustheit gegenüber Störungen führt, wird durch den Physical Layer gesteuert. Hierbei gibt der Master, der für das synchrone Hopping des Piconets verantwortlich ist, den Clients vor, welche Hopping-Sequenz verwendet werden soll. Auf diese Weise können verrauschte Frequenzbereiche umgangen werden. Der PHY verschickt die Datenpakete bei der Funkübertragung in Zeit-Slots, die eine unterschiedliche Länge haben können. Wie bei Ethernet befindet sich am Anfang der Pakete eine spezielle Sequenz, der sogenannte *Access Code*, der den Beginn eines Pakets andeutet.

Das **Baseband** ist für die Codierung und Modulation der Daten auf die Funkschnittstelle, das sogenannte *Framing*, verantwortlich. Dabei ist zu beachten, dass Bluetooth zwei Typen an Netzwerkverbindungen mit unterschiedlichen Paketformaten erlaubt. Den ersten Typ stellt der *Synchronous Connection-Oriented Link* (SCO) dar, der eine leitungsvermittelte Verbindung mit bis zu 64 kbit/s bereitstellt und zur Sprachübertragung genutzt wird. Der zweite Verbindungstyp ist der *Asynchronous Connectionless Link* (ACL), der eine paketvermittelte Übertragung für die restlichen Daten ermöglicht. Weiterhin enthält das Baseband Funktionen für den Aufbau und den Erhalt einer Verbindung, einen Master-Slave-Rollentausch und die Suche anderer in der Nähe befindlicher Bluetooth-Geräte.

Der **Link Manager** ist die oberste Hardwareschicht des Bluetooth-Controllers. Er ist für die Einrichtung, Konfiguration und Aufrechterhaltung von Verbindungen zuständig und nutzt dabei ein Zustandsmodell, das folgende Zustände für ein Gerät definiert:

- Inquiry: Gerät sucht nach unbekannten Bluetooth-Geräten in der Umgebung

- Inquiry Scan: Gerät hält Ausschau nach suchenden Geräten
- Page: Gerät baut Verbindung zu einem anderen Gerät auf
- Page Scan: Gerät erlaubt anderen Geräten sich mit ihm zu verbinden
- Connection-Active: aufgebaute ACL-Verbindung ist noch aktiv
- Standby: Gerät lauscht regelmäßig auf Inquiry- und Page-Anfragen

Der Link Manager ist aber auch für das Power-Management verantwortlich und nutzt spezielle ACL-Verbindungen, um einem Kommunikationspartner mitzuteilen, dass er seine Sendeleistung erhöhen muss oder verringern kann, um so möglichst energieeffizient zu arbeiten. Weiterhin ist er für das Einschalten der *Enhanced Data Rate* (EDR)-Übertragung, das Anstoßen des Master-Slave-Rollentausches sowie das optionale Aktivieren und Kontrollieren der Authentifizierung und Verschlüsselung zuständig.

Das auf dem Baseband aufsetzende **Logical Link Control and Adaptation Protocol (L2CAP)** regelt den Verbindungsaufbau und die Verschlüsselung der zu übertragenden Daten. Weiterhin realisiert es die Paketierung von Nachrichten und ist somit für die Aufteilung größerer Datenblöcke und das Zusammensetzen beim Empfänger verantwortlich. Damit dient es den höheren Anwendungsprotokollen als Datenmultiplexer und ermöglicht ihnen, Nachrichten (PDUs) bis zu einer Größe von 64 kbit auszutauschen. Auf Basis von ACL-Verbindungen ermöglicht das L2CAP zwar Verbindungen mit rudimentären QoS-Ansprüchen, benötigt eine Anwendung aber harte Echtzeitanforderungen, sollten eher SCO-Verbindungen mit konstanter Bitrate genutzt werden, die das Baseband direkt bereitstellt.

Auf dem L2CAP bauen mehrere Protokolle direkt auf. So dient z. B. das **Service Discovery Protocol (SDP)** dem Anbieten von Diensten auf dem lokalen Gerät und dem Suchen von Diensten auf benachbarten Geräten. Die Nutzung dieses Protokolls für eine Service-basierte Kommunikation wird im Abschnitt 2.3.2 noch genauer beschrieben. Das auch auf dem L2CAP aufbauende **Radio Frequency Communication (RFCOMM)**-Protokoll emuliert eine serielle Schnittstelle zwischen zwei Bluetooth-Geräten. Dazu stellt RFCOMM einen zuverlässigen Datentransfer mit Flusskontrolle bereit, der mehrere Verbindungen durch sogenannte *Channel Identifier* (CID) unterscheiden kann. Dieses Protokoll kann von den Anwendungen direkt für den Datenaustausch eingesetzt werden. Es bietet sich jedoch an, Daten als Objekte zu behandeln und unter Nutzung des **Object Exchange Protocols (OBEX)** mit anderen Geräten auszutauschen. OBEX arbeitet nach dem Client-Server-Prinzip und ist an das HTTP-Protokoll IP-basierter Netzwerke angelehnt. Es stellt Put- und

Get-Methoden für das Senden bzw. Empfangen von Daten bereit und bietet Anwendern eine unkomplizierte Möglichkeit, Dateien, Visitenkarten und Termindaten auszutauschen.

Ein weiteres Protokoll, das auf dem L2CAP aufbaut, ist das **Bluetooth Network Encapsulation Protocol (BNEP)**. BNEP ermöglicht eine direkte Kapselung und Übertragung von IP-basierten Daten. Damit stellt es für Bluetooth-Geräte die Grundlage der Nutzung IP-basierter Protokolle wie z. B. TCP und UDP dar und ermöglicht die Integration von Bluetooth-Geräten in IP-basierte Netzwerke. BNEP wird daher im Abschnitt 2.4.1 beim aktuellen Stand der Technik zur Interoperabilität IP-basierter und nicht-IP-basierter Netzwerktechnologien noch genauer vorgestellt.

Im Bluetooth-Standard werden Protokolle zu Profilen zusammengefasst, die einen bestimmten Anwendungsbereich beschreiben [35]. Sobald zwei Geräte das gleiche Profil unterstützen, ist gewährleistet, dass sie die gleichen Protokoll-Parameter nutzen und in diesem Anwendungsbereich zusammenarbeiten können. Für den Aufbau und die Nutzung einer Service-basierten Kommunikation durch das *Service Discovery Application Profile* (SDAP) müssen beispielsweise die Protokolle SDP und RFCOMM unterstützt werden. Das *Generic Access Profile* (GAP) beschreibt dagegen, wie sich Geräte angefangen vom Standby- bis zum Connected-Status verhalten und (sichere) Kommunikationskanäle zwischen ihnen aufzubauen sind. Weiterhin beschreibt das *Headset* (HS)-Profil die Verbindung eines Headsets mit einem mobilen Telefon. Bluetooth-Geräte können mehrere Profile gleichzeitig nutzen. Die Information, wer welche Profile unterstützt, kann zwischen den Bluetooth-Geräten jeweils nach dem Aufbau einer Verbindung ausgetauscht werden.

Adressierung in Bluetooth-Netzen

Für die Adressierung der Kommunikationspartner innerhalb eines Piconets verwendet der Bluetooth-Controller unterschiedliche Arten an Adressen. Die *BD_ADDR* ist eine 48 Bit lange Hardwareadresse, die der Bluetooth-Schnittstelle vom Hersteller fest vorgegeben ist und den MAC-Adressen der IP-basierten Netzwerktechnologien LAN und WLAN entspricht. Diese Adresse wird in einen 24 Bit langen *Lower Address Part* (LAP), einen 8 Bit *Upper Address Part* (UAP) und einen 16 Bit *Non-significant Address Part* (NAP) unterteilt. Innerhalb eines Piconets wird jedem aktiven Teilnehmer eine 3 Bit lange *LT_ADDR*-Adresse zugeordnet. Diese Adresse ist nur solange gültig, wie ein Knoten aktiv ist und begrenzt die maximale Anzahl aktiver Knoten in einem Piconet auf 8 Teilnehmer. Für Knoten, die nur passiv am Piconet teilnehmen, wird hingegen eine 8 Bit lange *PM_ADDR*-Adresse genutzt, die nur solange gültig ist, wie sich der Teilnehmer im Standby-Modus befindet. Die Nutzung der

LT_ADDR- und PM_ADDR-Adressen verringert die Größe der Datenpakete innerhalb eines Piconets deutlich und reduziert so den Protokoll-Overhead während der Kommunikation.

Auf Basis dieses Adressierungsschemas wurden für Bluetooth drei Paketarten definiert. Informationen zum Verbindungsaufbau, zur Sicherheit oder zur Funkübertragung selbst werden zwischen dem Master und den Clients durch spezielle *Control Packets* ausgetauscht. Während zur Audioübertragung SCO-Pakete genutzt werden, dienen ACL-Pakete der Best-Effort-Datenübertragung der Mehrzahl der höheren Protokolle. Dabei nutzt Bluetooth ein generelles Datenformat [31].

2.3.2 Service-basierte Kommunikation in Bluetooth

Die Bluetooth-Technologie definiert mit dem *Service Discovery Application Profile* (SDAP) die Protokolle, die ein Anbieten, Suchen und Nutzen von Diensten im Sinne einer Service-orientierten Architektur ermöglichen. Wie schon in den vorherigen Abschnitten kurz erwähnt, werden hierbei die Protokolle SDP und RFCOMM eingesetzt.

Das SDP-Protokoll stellt gegenüber den SOA-Technologien der IP-basierten Netze (wie z. B. Web Services) ein eher einfaches Protokoll zur Dienstbereitstellung und Dienstsuche dar. Es basiert auf dem Client-Server-Modell, wobei der SDP-Server einen Dienst auf dem lokalen Gerät bereitstellt und der SDP-Client Dienste auf benachbarten Geräten sucht. Ein separater Broker wird in Bluetooth-Netzen nicht verwendet, vielmehr übernimmt jeder SDP-Server die Broker-Funktionalität für die von ihm angebotenen Dienste. Dazu enthält ein SDP-Server ein Verzeichnis, die sogenannte *Service Discovery Database* (SDDB), in das Beschreibungen aller lokal angebotenen Dienste in Form von *Service Records* (SR) eingetragen werden. Jeder dieser Service Records besteht aus einer Menge an Dienstattributen. Dazu gehört die *ServiceID*, in der ein *Universally Unique Identifier* (UUID) [67] hinterlegt ist, der den Diensttyp eindeutig beschreibt. Üblicherweise bestehen UUIDs aus einer 128 Bit langen Nummer, die aus dem aktuellen Datum, der Uhrzeit und der MAC-Adresse der Bluetooth-Netzwerkkarte generiert wird. Diese einfache Generierung garantiert die Einzigartigkeit von UUIDs ohne die Nutzung einer zentralen Registrierung, denn entweder wurden die UUIDs auf dem selben Computer nacheinander zu unterschiedlichen Zeiten oder auf verschiedenen Geräten erstellt. Für Standard-Dienste wie OBEX können aber auch 16 Bit UUIDs verwendet werden. Derartige UUIDs basieren dann auf einer im Standard festgelegten 128 Bit Basis-UUID (00000000-0000-1000-8000-00805F9B34FB), deren höchstwertige 16 Bits je nach Diensttyp, Service-Klasse und genutztem Protokoll variabel gesetzt werden. Diese 16 Bit werden auch Alias-Adresse

bzw. 16 Bit-UUID genannt. Innerhalb der Service Records werden die 16 Bit-UUIDs für die Definition von Klassen genutzt, die beschreiben, welche Attribute ein Dienst zur Verfügung stellt. Baut z. B. ein Dienst auf der SerialPort-Klasse auf, so ist diese in seinem Attribut *ServiceClassIDList* vermerkt. Weiterhin enthält das Attribut *ProtocolDescriptorList* das vom Dienst unterstützte Transportprotokoll. Das Dienstverzeichnis unterstützt nicht nur die Dienstsuche anhand einer Dienstklasse oder ausgewählter Dienstattribute (UUIDs), sondern auch ein Service-Browsing, um nach allen Diensten eines Gerätes zu suchen.

Eine Dienstsuche beginnt in Bluetooth-Netzen üblicherweise mit der Suche der benachbarten Geräte durch einen Inquiry. Nachdem die Beantwortung des Inquirys die MAC-Adressen und Namen benachbarter Geräte liefert, kann jeder Nachbar anhand der MAC-Adresse direkt nach Diensten befragt werden. Dazu erstellt der SDP-Client ein sogenanntes Attribut-Set, d. h. eine Menge von Attributen, das er an den SDP-Server schickt. Der Server vergleicht die im Attribut-Set angegebenen UUIDs mit den Service Records in seinem Dienstverzeichnis und liefert standardmäßig die Werte der ersten fünf Dienstattribute (ServiceRecordHandle, ServiceClassIDList, ServiceRecordState, ServiceID, ProtocolDescriptorList) an den Client zurück, wenn ein Dienst als passend gilt. Da die MAC-Adresse des Servers bekannt ist und der zu nutzende RFCOMM-Kanal über den ServiceRecordHandle abgefragt werden kann, stehen dem SDP-Client alle Informationen zur Verfügung, die er für den Verbindungsaufbau zum Dienst benötigt. Die Methoden, wie ein Dienst genau zu nutzen ist, wird mit SDP jedoch nicht definiert und ist somit dienstabhängig.

2.3.3 Adressbasierte Kommunikation in ZigBee

Der ZigBee Protokollstack

Die unterschiedlichen Aufgaben, die für eine standardisierte Adressierung und Kommunikation in einem Netzwerk bewältigt werden müssen, sind auch in ZigBee-basierten Netzen durch einen Protokollstack festgelegt. Dieser Protokollstack ist, wie in Abbildung 2.13 dargestellt, in verschiedene Schichten unterteilt, deren Funktion und Zusammenarbeit in den folgenden Abschnitten kurz erläutert wird.

Der **Physical Layer** ist im Standard 802.15.4 [68] definiert und stellt die unterste Schicht des ZigBee-Protokollstacks dar. Gegenüber dem PHY des Bluetooth-Stacks gibt es für ZigBee zwei unterschiedliche PHYs, die sich im verwendeten Frequenzbereich und der Datenmodulation unterscheiden. So können zum einen im 2,4 GHz Frequenzbereich des ISM-Bandes 16 Kanäle mit einer Datenrate von bis zu 250 kbit/s

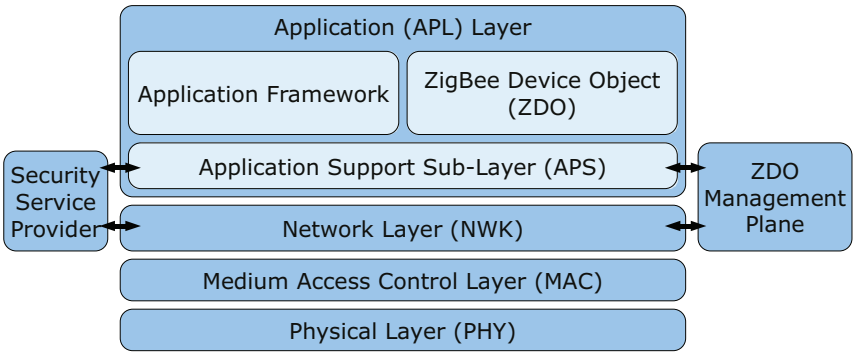

Abbildung 2.13 Aufbau des ZigBee-Stacks

(*Offset Quadrature Phase Shift Keying* (OQPSK)) weltweit genutzt werden. Zum anderen wird regional abhängig, in Amerika der 915 MHz-Bereich für 10 Kanäle mit bis zu 40 kbit/s und in Europa ein Kanal im 868 MHz-Bereich mit einer Datenrate von 20 kbit/s verwendet (*Binary Phase Shift Keying* (BPSK)). Wie bei Bluetooth erlaubt auch der ZigBee-PHY eine grobe Pegelmessung in Form eines RSSI-Indikators, der ein Maß für die Funkqualität zwischen zwei ZigBee-Geräten angibt.

Der auf dem PHY-Layer aufbauende **Medium Access Control Layer (MAC-Layer)** wird ebenfalls durch den Standard 802.15.4 definiert. Er regelt den Zugriff auf die Funkhardware und nutzt dabei ein CSMA/CA-Verfahren, das je nach Netzwerktopologie Slot-basiert und durch einen Koordinator kontrolliert oder aber auch zufällig in einem Ad-hoc-Modus durchgeführt werden kann. Wird ein Koordinator im ZigBee-Netz eingesetzt, besteht eine wesentliche Aufgabe seines MAC-Layers in der Übertragung sogenannter *Beacon Frames*, die der Synchronisation des Netzes dienen. Um den Koordinator des Netzes zu ermitteln, kann ein ZigBee-Knoten entweder passiv auf so einen Beacon warten oder explizit durch einen *Beacon Request* die Existenz des Koordinators erfragen. Anschließend bekommt der Knoten eine 16 Bit lange *Short-Adresse* als Antwort zurück, die der Piconet-Nummer eines Bluetooth-Gerätes ähnelt und nur innerhalb des ZigBee-Netzes gültig ist. Auch bei ZigBee werden MAC-Adressen und zughörige Short-Adressen beim Koordinator in einer Tabelle hinterlegt. Auf den Vorgang der Adresszuweisung und -nutzung wird am Ende des Abschnitts 2.3.3 noch genauer eingegangen.

Auf Basis der durch den Standard 802.15.4 definierten PHY- und MAC-Layer baut die ZigBee Alliance [33] einen **Network Layer (NWK)** auf, der eine Stern-, Baum- und Mesh-Topologie des Netzwerks unterstützt. Innerhalb der Topologie übernehmen ZigBee-Knoten unterschiedliche Rollen. So implementiert ein einfacher Knoten als *Reduced Function Device* (RFD) nur einen Teil der ZigBee-Protokolle und bildet mit einem Router als *Full Function Device* (FFD) ein Netzwerk in Sterntopologie.

FFDs melden sich wiederum an existierenden FFDs im Netz an und bilden ein Netzwerk in Baum-Topologie, in dem ein FFD zusätzlich die Rolle des Koordinators übernimmt und ähnlich einem Master in einem Bluetooth-Netz das Netzwerk koordiniert. Hierbei wird auf die Beacon-orientierte Kommunikation des MAC-Layers zurückgegriffen. Durch Ausnutzung von Abkürzungen kann aus der Baum-Topologie eine Mesh-Topologie entstehen.

Der **Application Layer (APL)** des ZigBee-Protokollstacks entspricht der Anwendungsschicht des TCP/IP-Referenzmodells in IP-basierten Netzwerken. Sein unterster Teil, der *Application Support Sublayer* (APS) dient als Schnittstelle zum Network Layer und stellt ein Set an generellen Funktionen bereit, die vom *Application Framework* und vom *ZigBee Device Object* (ZDO) benutzt werden. Dazu gehören z. B. Funktionen, die dem Binding und der Sicherheit dienen. Weiterhin führt er das Konzept von Endpoints ein, das dem Konzept der Bluetooth-Kanäle entspricht und die eindeutige Zuordnung von Daten zu einer Anwendung ermöglicht. Im Application Framework werden bis zu 240 sogenannte Application Objects gehostet, die durch einen Endpoint identifiziert werden können. Der Endpoint 255 ist für Daten vorgesehen, die als Broadcast an alle Anwendungen weitergeleitet werden sollen. Die Endpoints 241 bis 254 sind für zukünftige Erweiterungen des Protokollstacks reserviert. Der Endpoint 0 ist für das ZDO-Interface reserviert, das gemeinsame Ansprüche aller Anwendungen erfüllt. So ist es z. B. für die Initialisierung des APS, des NWKs und des Security Service Providers zuständig und legt damit Konfigurationsinformationen für alle End-Anwendungen fest.

Eine besondere Bedeutung kommt der **ZDO Management Plane** zu, in der Informationen zum Typ und zu den Fähigkeiten eines Knotens, zu Energiecharakteristiken und Gerätebeschreibungen sowie Informationen über die genutzten Endpoints, als Deskriptoren in einer Tabelle gespeichert und von anderen Geräten durch eine Anfrage an das ZDO (Endpoint 0) abgefragt werden können. Auf diese Weise enthält die ZDO Management Plane für jede Anwendung im Application Framework einen Deskriptor, der für die Dienstsuche und -nutzung im Sinne einer Service-orientierten Architektur verwendet werden kann.

Adressierung in ZigBee-Netzen

ZigBee verwendet unterschiedliche Adressierungsarten für die Kommunikation zwischen zwei Anwendungen. Während bei der direkten Adressierung die Anwendung der Gegenstelle mit Hilfe einer 64 Bit MAC-Adresse und der Nummer des 8 Bit langen Endpoints eindeutig adressiert wird, nutzt die indirekte Adressierung statt der MAC-Adresse die vom Koordinator des Netzwerks vergebene 16 Bit Short-Adresse. Der Ko-

ordinator speichert das Paar aus MAC- und Short-Adresse in einer Binding-Tabelle. Dieses Adresspaar können andere ZigBee-Teilnehmer in Erfahrung bringen, indem sie entweder die Short-Adresse eines Gerätes kennen und eine Anfrage zur Bestimmung der MAC-Adresse als Unicast an den Koordinator stellen oder einen Broadcast, der die bekannte MAC-Adresse enthält, als Anfrage zur Bestimmung der Short-Adresse durchführen. Die Nutzung von Short-Adressen ermöglicht nicht nur deutlich kürzere Datenpakete, sondern auch den Einsatz spezieller Gruppen-Adressen, mit denen Nachrichten an eine Short-Adresse an mehrere ZigBee-Geräte weitergeleitet werden können. Als Einsatzbeispiel sei hier ein Lichtschalter zu nennen, der beim Verlassen der Wohnung alle Lampen gleichzeitig ausschaltet. Möchte ein Knoten mit einem anderen kommunizieren, ist es so zwar notwendig zuerst eine Anfrage an den Koordinator zu senden, der sie an den jeweiligen Empfängerknoten weiterleitet, das Verwalten des Netzwerks wird dadurch aber erheblich vereinfacht und flexibilisiert. Der Ausfall des Koordinators kann bei Verwendung indirekter Adressierung den Ausfall des gesamten Netzes bedeuten, da dieser alle Routing- und Geräteinformationen in einem lokalen Speicher hält. Daher ist es ratsam, weitere FFDs so zu konfigurieren, dass sie im Fehlerfall die Aufgabe des Koordinators übernehmen.

Auf Basis dieser Adressierungsarten wurden für ZigBee zwei Paketarten definiert, die als *NWK Command Frame* bzw. *Data Frame* auf einem allgemeinen Paketformat basieren und sowohl der Koordination der Netzwerktopologie als auch dem Datenaustausch zwischen den ZigBee-Geräten dienen [34].

2.3.4 Service-basierte Kommunikation in ZigBee

Auch in ZigBee-Netzen besteht die Möglichkeit, die Fähigkeiten eines Gerätes anderen Netzwerkteilnehmern in Form von Diensten bereitzustellen. Die Informationen zu den Fähigkeiten eines Knotens sowie Informationen über die genutzten Endpoints werden hierzu durch das ZDO als Deskriptor in einer Tabelle in der ZDO Management Plane gespeichert, die von anderen Geräten durch eine Anfrage an das ZDO (Endpoint 0) abgefragt werden können. Damit enthält die ZDO Management Plane für jede Anwendung im Application Framework einen Deskriptor, der für die Dienstsuche im Sinne einer Service-orientierten Architektur eingesetzt werden kann.

Die Dienstsuche kann durch unterschiedliche Anfragen realisiert werden. So kann eine Anfrage als Unicast an die ZDO eines bekannten Gerätes gestellt werden. Ebenfalls ist auch eine Anfrage als Broadcast an das gesamte Netzwerk möglich. In beiden Fällen sind Kriterien in der Anfrage enthalten, die der gewünschte Dienst erfüllen muss. Wird ein passender Dienst angeboten, antwortet das entsprechende Gerät mit

einer Unicast-Nachricht. Um Energie zu sparen, ist es in ZigBee-Netzen auch möglich, dass Koordinatoren die Dienstinformationen einzelner Teilnehmer cachen und stellvertretend auf die Anfrage der Dienstsuche antworten. Damit stehen auch hier die Adresse des Dienstanbieters und des Endpoints, unter denen der Dienst erreichbar ist, für eine Verbindung zum Dienst und damit zur eigentlichen Dienstnutzung bereit.

2.4 Stand der Forschung zur Interoperabilität heterogener Netzwerktechnologien

Nachdem die vorherigen Abschnitte einen tieferen Einblick in die Grundlagen der Kommunikation aktueller IP-basierter und nicht-IP-basierter Netzwerktechnologien gegeben haben, widmen sich die folgenden Abschnitte dem aktuellen Stand der Forschung zur Interoperabilität der LAN-, WLAN- und WPAN-Technologien. Hierbei gibt es unterschiedliche Ansätze, die sich grob in die in Abbildung 2.14 dargestellte Internet- bzw. Anwendungsschicht des TCP/IP-Referenzmodells einordnen lassen.

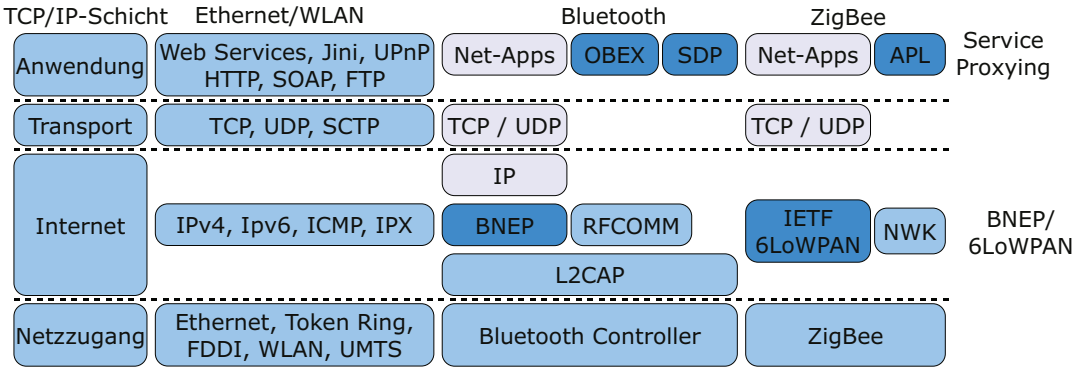

Abbildung 2.14 Interoperabilität auf der Internet- und der Anwendungsschicht des TCP/IP-Modells

Auf der Ebene der Internetschicht verfolgen aktuelle wissenschaftliche Arbeiten das Ziel einer All-over-IP-basierten Kommunikation. Dabei werden Erweiterungen des Bluetooth-Stacks (BNEP) und des ZigBee-Stacks (6LoWPAN) eingesetzt, die die Nutzung von IP-Adressen und darauf aufbauender TCP- und UDP-Verbindungen in WPANs ermöglichen. Da aber Dienste zunehmend direkt auf mobilen Sensoren angeboten werden und die Unabhängigkeit von Infrastruktur und benötigter IP-Adressen als Voraussetzung der Kommunikation eigentlich ein großes Ziel bei der Entwicklung von Bluetooth und ZigBee war, basieren andere Ansätze nicht auf dem All-over-IP-Ansatz, sondern nutzen zur Interoperabilität auf der Anwendungsschicht

sogenannte Service-Proxys, die Dienste einer Netzwerktechnologie in einer jeweils anderen auffindbar und nutzbar machen. Die aktuellen Ansätze zur Interoperabilität verschiedener Netzwerk-Technologien werden in den nun folgenden Abschnitten näher erläutert, bevor aufgezeigt wird, wie sich das Thema der vorliegenden Arbeit vom aktuellen Stand der Technik abhebt.

2.4.1 Interoperabilität auf der Internetschicht

Bluetooth Network Encapsulation Protocol

Für die Integration von Bluetooth in IP-basierte Netzwerke wie z. B. WLAN und Ethernet, hat die Bluetooth-SIG ein spezielles Protokoll im Bluetooth-Stack vorgesehen. Dieses *Bluetooth Network Encapsulation Protocol* (BNEP) [69] erweitert den klassischen Bluetooth-Stack um eine Kapselung von Ethernet-Paketen in Bluetooth-Pakete und ermöglicht eine direkte Übertragung von unveränderten IP-basierten Daten über Bluetooth-Verbindungen. BNEP ist dabei für den Transport von Kontroll- und Daten-Paketen zuständig. Wie schon in Abbildung 2.12 dargestellt, setzt BNEP auf L2CAP-Verbindungen des Bluetooth-Stacks auf. Darunterliegende Bluetooth-Schichten werden dabei als Übertragungsmedium aufgefasst und in die Netzzugangsschicht des TCP/IP-Referenzmodells neben Ethernet und WLAN eingeordnet. Durch die auf BNEP aufbauende IP-Schicht können nun die aus IP-basierten Netzen gewohnten Internetprotokolle IPv4 und IPv6 sowie die darauf aufbauenden IP-basierten Transportprotokolle UDP und TCP genutzt werden. Dies führt zur Interoperabilität zwischen Bluetooth und IP-basierten Netzwerken auf der Ebene der Internetschicht des TCP/IP-Referenzmodells.

Damit IP-basierte Datenpakete mit Hilfe von L2CAP-Verbindungen übertragen werden können, wird am Anfang des L2CAP-Payloads ein BNEP-Header eingefügt, der je nach Art des BNEP-Paketes eine Länge von 3 oder 15 Bytes hat. Das generelle BNEP-Headerformat ist 15 Bytes lang und enthält ein Feld für den Typ des Headers, jeweils 6 Byte lange Destination- und Source-MAC-Adressen sowie den *Network Protocol Type*, der auch in Ethernet-Frames eingesetzt wird. Das *Compressed*-Headerformat ist deutlich kürzer, da es nur dann eingesetzt wird, wenn sich Sender und Empfänger im gleichen Bluetooth-Piconet befinden und somit auch über ihre Short-Adressen eindeutig adressierbar sind. Auf diese Weise können die zu übertragenden Daten auch unter Nutzung von BNEP reduziert werden. Nach dem BNEP-Header folgen die eigentlich zu übertragenden Daten, die als Ethernet-Payload aus dem Ethernet-Paket herausgetrennt und unverändert in das L2CAP-Paket eingefügt werden. Auf diese Weise kapselt L2CAP den BNEP-Header sowie den Ethernet-Payload ein und überträgt beide über die Bluetooth-Schnittstelle.

Für die Übertragung definiert BNEP verschiedene Rollen für Bluetooth-Geräte [70], von denen besonders die Rollen der in Abbildung 2.15 dargestellten *Personal Area Network User* (PANU) und des *Network Access Points* (NAP) für das Konzept der vorliegenden Arbeit von Bedeutung sind.

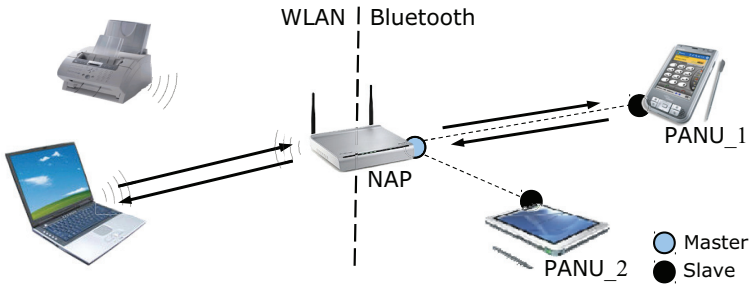

Abbildung 2.15 Rollenverteilung bei BNEP am Beispiel der Datenvermittlung zwischen einem WLAN und einem Bluetooth-Netzwerk

Bei dem PANU handelt es sich um einen normalen Bluetooth-Slave, der nur über eine Bluetooth-Schnittstelle verfügt. Der NAP in diesem Beispiel ist dagegen mit einer WLAN- und einer Bluetooth-Schnittstelle ausgestattet und dient als Vermittler zwischen dem Bluetooth-Netzwerk und dem WLAN. Bei der Kommunikation zwischen dem Laptop und dem PDA (PANU_1) werden die IP-basierten Daten an den NAP gesendet. Dort wird der Ethernet-Payload der Pakete durch BNEP herausgetrennt, in L2CAP-Pakete gekapselt und an den PDA weitergeleitet. Der PDA packt die Pakete unter Nutzung von BNEP wieder aus und übergibt den Payload an die höheren Schichten. Eine darauf folgende Antwort wird durch den PDA wieder in ein L2CAP-Paket gekapselt und an den NAP zurück gesendet, der den Payload in Ethernet-Frames einbettet und an den Laptop weiterleitet.

BNEP wird in verschiedenen Forschungsaktivitäten für die Integration von Bluetooth in IP-basierte Netzwerke eingesetzt. Dazu zählt zum Beispiel *BlueStar* [71], das den mit BNEP einhergehenden All-over-IP-Ansatz für den Zugang mobiler Geräte zum Internet nutzt. In Anlehnung an einen Bluetooth-NAP setzt BlueStar sogenannte *Bluetooth Wireless Gateways* (BWGs) ein, die mit einer WLAN- und einer Bluetooth-Schnittstelle ausgestattet sind und als Zugangspunkt zur globalen Infrastruktur des Internets dienen.

Weiterhin wird BNEP in [72] für eine nahtlose Nutzung heterogener IP-basierter Netzwerkschnittstellen eingesetzt. Dabei werden Geräte genutzt, die gleichzeitig über Ethernet-, Bluetooth- und WLAN-Schnittstellen verfügen und je nach benötigter Datenrate und Reichweite mit Hinblick auf einen möglichst geringen Energieverbrauch bei der Übertragung die gerade zu verwendende Netzwerkschnittstelle auswählen.

Die Ergebnisse dieses Ansatzes zeigen, dass durch den Einsatz mehrerer unterschiedlicher Netzwerkschnittstellen die Konnektivität, Erreichbarkeit und Zuverlässigkeit der Netzwerkteilnehmer erhöht werden kann. Die eigentlich zu lösenden Probleme bei ihrer Betrachtung heterogener, All-over-IP-basierter Netzwerke liegen aber eher im Bereich geeigneter Metriken für Routing-Protokolle, die die Heterogenität der Netzwerkschnittstellen in Bezug auf Latenz, Reichweite, Datenrate und Energieverbrauch berücksichtigen können.

BNEP wird aber auch zur Multicast-Übertragung von Echtzeit-Audio in Bluetooth-Netzen verwendet [73]. Dabei werden Audio-Daten mit deutlich höherer Datenrate zwischen den Netzwerkteilnehmern per UDP-Verbindung ausgetauscht, als es die durch Bluetooth bereitgestellten Unicast-SCO-Verbindungen mit ihrer maximalen Datenrate von 64 kbit/s ermöglichen. Hierbei wird BNEP speziell wegen der Bereitstellung von Multicasts eingesetzt, die SCO-Verbindungen nicht bieten. Bei den Messergebnissen zeigt auch dieser Ansatz, dass sich der Overhead durch die Nutzung von BNEP negativ auf die Übertragungsdatenrate auswirkt.

Weitere Arbeiten die BNEP einsetzen, beschäftigen sich z. B. mit der theoretischen und praktischen Untersuchung des Einflusses von BNEP auf die Latenz in Bluetooth-Netzen [74], was aber nicht im Fokus dieser Arbeit liegt.

BNEP bietet einige Vorteile für die Interoperabilität von Bluetooth und IP-basierten Netzen wie z. B. WLAN und LAN. Dazu gehört die Bereitstellung einer All-over-IP-basierten Kommunikation für die Bluetooth-Protokolle oberhalb von BNEP. Damit ist es für die höheren Protokolle ab der Internetschicht des TCP/IP-Modells transparent, welche Netzwerktechnologie zur Datenübertragung mit einem Kommunikationspartner eingesetzt wird. Auf diese Weise können Protokolle wie z. B. HTTP, SOAP und FTP nun auch auf Bluetooth-basierten Geräten eingesetzt werden.

BNEP hat aber gerade bei der Kommunikation zwischen Bluetooth-Geräten auch Nachteile. So führt der Einsatz von BNEP dazu, dass auf jedem Bluetooth-Gerät ein IP-Stack vorhanden sein muss, der gerade für Sensoren als Teil einer pervasiven Umgebung zu einem erheblichen Software-Overhead sowohl auf dem Sensor selbst als auch während der Datenübertragung führt. Weiterhin bedeutet die Nutzung von BNEP, dass nun in Bluetooth-Netzen Mechanismen aus den IP-basierten Netzen benötigt werden, mit denen jedem Gerät eine im lokalen Netzwerk eindeutige IP-Adresse sowie die Adressen der Gateway- und Nameserver zugewiesen werden können. Diese werden für die Kommunikation im reinen Bluetooth-Netz jedoch nicht benötigt. Der NAP des Bluetooth-Netzes ist zusätzlich für die Realisierung von NAT zuständig (siehe Abschnitt 2.2.2), was zur Einschränkung der im Bluetooth-Netz ansprechbaren Ports führt. Weiterhin führt der Einsatz IP-basierter Kommunikation dazu, dass nun IP-basierte Service-Technologien wie z. B. Web Services statt SDP zur

Kommunikation zwischen den Bluetooth-Geräten eingesetzt werden, die zum einen Anpassungen an vorhandenen Anwendungen benötigen und zum anderen einen deutlichen Protokoll-Overhead während der Kommunikation verursachen, worauf später in dieser Arbeit noch detaillierter eingegangen wird.

Zusammenfassend bietet BNEP eine Integration von Bluetooth in das IP-basierte Kernnetz des Internets, die jedoch mit erheblichem Management- und Protokoll-Overhead verbunden ist, der für eine reine Kommunikation zwischen Bluetooth-Geräten nicht notwendig ist.

IPv6 over Low power WPAN

Unter dem Namen *IPv6 over Low power WPAN* (6LoWPAN) befasst sich seit einiger Zeit eine Arbeitsgruppe der IETF [75] mit einer Spezifikation, die den Einsatz von IPv6 in ZigBee-Netzwerken und somit eine Integration in IP-basierte Netzwerke mit möglichst geringem Aufwand ermöglichen soll. Aufbauend auf dem Standard 802.15.4-2006 wurden dazu mit RFC 4919 [76] und RFC 4944 [77] bereits zwei Erweiterungen spezifiziert, von denen der RFC 4919 sowohl einen allgemeinen Überblick gibt als auch die Ziele und Probleme bei der Integration von IPv6 in WPANs beschreibt. Der RFC 4944 geht dagegen tiefer auf die Adressierung und die zu verwendenden Header-Formate ein. Zukunftsweisend ist hier der Einsatz von IPv6, das aufgrund der verwendeten 128 Bit-Adressen gegenüber IPv4 einen deutlich größeren Adressraum besitzt und der erwarteten stark zunehmenden Anzahl an ZigBee-Sensoren in den nächsten Jahren gerecht werden kann. Aber auch IPv6-Mechanismen, die eine Autokonfiguration der Netzwerkteilnehmer erleichtern oder durch Header-Kompression die Paketgröße vor der Übertragung verringern, haben zur Entscheidung für den Einsatz von IPv6 auf ressourcenarmen Geräten beigetragen.

Bisherige Forschungsarbeiten betrachten zwar 6LoWPAN und IPv6-basierte Sensornetze als nächsten großen Schritt für die Interoperabilität von ZigBee und IP-basierten Netzen, sie entwickeln jedoch teilweise eigene IPv6-Stacks für ZigBee-Sensoren [78] bzw. eigene Gateway-Designs [79], die eine Ende-zu-Ende-Interoperabilität ermöglichen. Auf diese Weise können z. B. im Bereich des Katastrophenmanagements Sensoren ihre Daten direkt an zentrale Managementstationen im Internet übertragen, ohne umständliche Proxy-Lösungen beim Übergang zwischen nicht-IP-basierten ZigBee-Netzen und dem IP-basierten Internet nutzen zu müssen [80].

Weitere Arbeiten [81][82], die 6LoWPAN für ihre Forschungsaktivitäten einsetzen, fokussieren eher die eingebaute Unterstützung der AES-128-Bit-Verschlüsselung als Basis einer robusten Authentifizierung und sicheren Datenübertragung zwischen ZigBee-Sensoren und Geräten in klassischen WLANs und LANs.

2.4.2 Interoperabilität auf der Anwendungsschicht

Ein anderer Ansatz zur Interoperabilität von nicht-IP-basierten WPANs und dem IP-basierten Kernnetz des Internets verbindet die unterschiedlichen Netzwerktechnologien auf der in Abbildung 2.14 dargestellten Anwendungsschicht. Dieser Ansatz nutzt die bereits vorgestellten Methoden der Service-basierten Kommunikation der einzelnen Netzwerktechnologien und verbindet sie vorrangig durch einen Proxy, der es ermöglicht, Dienste eines fremden Netzwerks finden und nutzen zu können.

So stellt z.B. [83] eine Architektur vor, die die Service-Technologien SDP (Bluetooth) und Jini (TCP/IP) zusammenführt. Dazu wurde ein Jini-Profil für Bluetooth entwickelt und auf den Bluetooth-Geräten verwendet, das als *Surrogate* oder *Client* arbeiten kann. Als Surrogate wird das Profil auf einem Gerät genutzt, das mit Bluetooth- und Ethernet-Schnittstelle ausgestattet ist und zwischen beiden Netzwerktechnologien vermittelt. Es bietet sich durch einen SDP-Dienst im Bluetooth-Netz als Surrogate an. Mit Hilfe des Jini-Profils führt der Bluetooth-Client in Jini verwendete *Service Lookup Requests* an das Surrogate durch. Das Surrogate parst diese Anfragen, die als XML-Dokumente durch das OBEX-FTP-Profil an das Surrogate übertragen werden, konvertiert sie in Jini *Service Templates* und bietet sie im IP-basierten Netzwerk an. Auf diese Weise wird die Nutzung der Service-Technologie Jini in das Bluetooth-Netzwerk hinein ausgedehnt.

Auch für die Service-Technologie UPnP gibt es einen Ansatz [84], der die Reichweite der Service-Technologie in nicht-IP-basierte Netze ausdehnt. Dazu wurde eine Architektur für einen Proxy-Server entwickelt, der UPnP-Dienste vom Bluetooth-Netzwerk aus auffindbar und nutzbar macht. Möchte ein Bluetooth-Client also einen UPnP-Dienst nutzen, so sendet er eine *Invocation*-Nachricht an den Proxy-Server, der die Nachricht neu formatiert und an die UPnP-Umgebung weiterleitet. Somit dient der Proxy-Server als Vermittler, indem er Anfragen von Clients ähnlich einem Web-Proxy an andere Server weiterleitet. Da die Bluetooth-Clients nach Diensten einer eigentlich fremden Service-Technologie suchen, wird zwar die Nutzung der Service-Technologie UPnP in das Bluetooth-Netz ausgedehnt, die Nutzung der speziell an Bluetooth angepassten Service-Technologie SDP geht jedoch auch hier verloren.

Ein anderer Ansatz [85] fokussiert die Nutzung von Web Services durch mobile Bluetooth-basierte Geräte. Hierbei wird analysiert, wie entfernte Web Services unter Nutzung von RFCOMM aus dem Bluetooth-Netz heraus aufgerufen und genutzt werden können. Auch dieser Ansatz versucht, eine IP-basierte Service-Technologie in einem nicht-IP-basierten WPAN zu nutzen, ohne auf die dort vorhandene Service-Technologie SDP zurückzugreifen.

Dagegen befasst sich [86] mit der Erhöhung der Reichweite der in Bluetooth-Netzen eingesetzten Service-Technologie SDP und nutzt spezielle *Hot-Spots* als Service-Proxys, die SDP-Dienste zwischen entfernten Bluetooth-Netzen über ein IP-basiertes Backbone tunneln und Dienste eines Bluetooth-Netzes in einem räumlich entfernten Bluetooth-Netz zur Verfügung stellen. Ein einfaches Beispiel dieses Ansatzes ist in Abbildung 2.16 dargestellt.

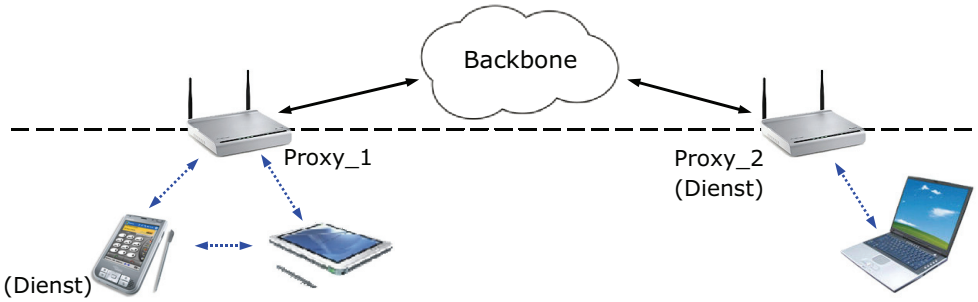

Abbildung 2.16 Tunneln von Bluetooth SDP-Diensten über ein Ethernet-basiertes Backbone

Hierbei werden zwei Service-Proxys als Vermittler eingesetzt. Damit der vom PDA angebotene SDP-Dienst vom Laptop im entfernten Bluetooth-Netz gefunden und genutzt werden kann, sucht Proxy_1 zunächst nach vorhandenen Diensten der lokal erreichbaren Bluetooth-Geräte (vgl. Abschnitt 2.3.2) und teilt Proxy_2 den gefundenen Dienst mit, der ihn stellvertretend für den Laptop anbietet. Durch eine lokale Dienstsuche kann der Laptop dann den stellvertretenden Dienst finden. Für die Nutzung des Dienstes werden die zwischen dem PDA und dem Laptop auszutauschenden Daten durch die Proxys über das IP-Netz weitergeleitet. Damit erweitern die Proxys die Service-basierte Kommunikation auf transparente Weise über die Grenze des lokalen Bluetooth-Netzes hinaus. Hierbei tunneln sie die Daten aber nur und führen keine Umsetzung in eine andere Service-Technologie durch. Bei diesem Ansatz wird zwar die Service-Technologie des WPANs berücksichtigt und in ihrer Reichweite in entfernte Bluetooth-Netze ausgedehnt, die Interoperabilität zwischen unterschiedlichen IP- und nicht-IP-basierten Service-Technologien betrachtet dieser Ansatz aber nicht.

Auch der ZigBee-Technologie widmen sich verschiedene Forschungsaktivitäten der Interoperabilität mit IP-basierten Service-Technologien. So beschreibt [87] die Kombination von ZigBee-Sensoren und *Home Gateways*, die eine dynamische Integration von ZigBee-basierten Geräten als Device-Proxy-Dienste in OSGi-basierte Netze ermöglichen. Sobald ein ZigBee-Gerät dem Netzwerk beitritt, wird dessen Dienstprofil genutzt, um einen stellvertretenden Proxy-Dienst in der OSGi-Service-Registry zu

registrieren, dessen OSGi-Bundle aus dem OSGi-basierten Netz heraus automatisch herunterzuladen, zu installieren und wie ein gewöhnliches OSGi-Gerät zu behandeln. Ein vergleichbarer Ansatz beruht auf UPnP-ZigBee-Gateways [88], die für jedes ZigBee-Gerät einen virtuellen UPnP-Proxy im IP-basierten Netzwerk erstellen. Weitere Ansätze im Bereich der Interoperabilität auf Anwendungsebene [89][90][91] setzen auf den Einsatz von 6LoWPAN im ZigBee-Netz und setzen damit eine IP-basierte Kommunikation voraus, auf dessen Basis bekannte IP-basierte Service-Technologien wie z. B. Web Services direkt genutzt werden können. Da dieser Ansatz jedoch zu starkem Protokoll-Overhead bei der Kommunikation führt, geht der Trend mit *Device Profile for Web Services* (DPWS) [92] in diesem Bereich in Richtung von Web Services, die auf die speziellen Bedürfnisse mobiler, eingebetteter, ressourcenarmer Systeme und das dynamische Entdecken von Diensten sowie die ereignisbasierte Kommunikation angepasst sind.

2.5 Abgrenzung dieser Arbeit zum Stand der Forschung

Der aktuelle Stand der Forschung bietet mit den vorgestellten Ansätzen zwei Bereiche zur Interoperabilität von WPAN-Technologien und dem IP-basierten Kernnetz des Internets. Der erste Bereich befasst sich mit der Interoperabilität auf der Internetschicht des TCP/IP-Referenzmodells und verwendet einen All-over-IP-basierten Ansatz. Dabei werden die Protokoll-Stacks der WPAN-Technologien um die IP-Funktionalität erweitert. Dieser Ansatz führt gerade auf kleinen, mobilen Geräten durch die nun benötigte Zuweisung von IP-, Nameserver- und DNS-Adressen zu einem nicht vernachlässigbaren Overhead bei der Konfiguration des Netzwerks. Aber gerade beim Datenaustausch zwischen benachbarten Bluetooth-Geräten (ebenso zwischen ZigBee-Geräten) im lokalen Netzwerk, die normalerweise ohne die Nutzung des IP-Stacks miteinander kommunizieren könnten, stellt der All-over-IP-Ansatz einen enormen Protokoll-Overhead während der Kommunikation dar.

Der zweite Bereich der vorgestellten Ansätze befasst sich mit der Interoperabilität von WPAN-Technologien und dem IP-basierten Kernnetz auf der Anwendungsschicht des TCP/IP-Referenzmodells und nutzt das Konzept des Service-Proxying. Der aktuelle Stand der Forschung zeigt, das die bisherigen Ansätze auf höchstens zwei Service-Technologien beschränkt sind und dabei versuchen, die Nutzung IP-basierter Service-Technologien in nicht-IP-basierte WPANs durch spezielle Profile auszudehnen, ohne auf die dort vorhandenen Service-Technologien zurückzugreifen. Ansätze, die ein Tunneln von Diensten zwischen entfernten WPANs gleicher Technologie ermöglichen, gibt es bisher nur für Bluetooth; die vorgestellten Service-basierten

Ansätze im Bereich der ZigBee-Technologie unterstützen diese Form der Kommunikation nicht. Auch eine Service-basierte Kommunikation zwischen unterschiedlichen WPAN-Technologien wurde bisher nicht betrachtet.

Intelligente, heterogene Umgebungen, so wie sie z. B. im MuSAMA-Projekt eingesetzt werden, fokussieren eine Informationsverarbeitung, die ubiquitär über Technologiegrenzen hinweg und mit Fokus auf das kooperative Verhalten der Geräte zur proaktiven Unterstützung des Nutzers in seiner Umgebung durchgeführt wird. Dabei ist es notwendig, dass alle Geräte innerhalb des Ensembles Technologie-übergreifend miteinander kommunizieren können, unabhängig davon mit welcher Art von Netzwerkschnittstelle sie ausgestattet sind. Dazu wird in der vorliegenden Arbeit ein allgemeineres Konzept entwickelt, das nicht nur die Kommunikation zwischen dem IP-basierten Netzwerk und den WPANs ermöglicht, sondern auch die Kommunikation zwischen unterschiedlichen WPAN-Technologien, die jeweils an das Internet angebunden sind. Ebenso wird ein transparentes Tunneln der Kommunikation zwischen entfernten WPANs unterstützt, die gleiche Service-Technologien einsetzen. Dabei wird ein geeignetes Modell zur Interoperabilität vorgeschlagen, das die Service-Technologien der WPANs berücksichtigt, jedoch nicht auf eine durchgängige IP-basierte Adressierung und Kommunikation angewiesen ist. Der Möglichkeit einer nahtlosen Integration neuer drahtgebundener und drahtloser Netzwerktechnologien in das heterogene Netzwerk kommt hierbei eine tragende Bedeutung zu, die den fortschreitenden Entwicklungen im Bereich der Netzwerktechnologien Rechnung trägt und einfache Erweiterungsmöglichkeiten bietet.

Kapitel 3

Referenzarchitektur zur allgegenwärtigen Kommunikation heterogener Systeme in Smart Ensembles

Als Grundlage der theoretischen Betrachtung dieser Arbeit wird in diesem Kapitel zuerst ein genauer Blick auf die aktuelle Kommunikation in heterogenen Netzen geworfen, wobei besonders auf horizontale und vertikale Netzwerkstrukturen und deren Kombination zu einem heterogenen Ensemble eingegangen wird. Darauf aufbauend wird eine theoretische Bestimmung der logischen Konnektivität eines Netzwerks beschrieben, die ein Maß für den Zusammenhalt eines Netzes widerspiegelt. Danach werden zwei mögliche Organisationsformen für Gateways vorgestellt, die als Vermittler zwischen den verschiedenen Netzwerktechnologien eingesetzt werden können. Anschließend wird ein Ansatz zur Service-basierten Kommunikation entwickelt, der in Kombination mit einem adressbasierten Kommunikationsansatz zwischen unterschiedlichen Netzwerktechnologien, der Mobilität von Endgeräten und deren Diensten Rechnung trägt und somit die Basis für die praktischen Betrachtungen im Kapitel 4 bietet. Darauf aufbauend wird die Referenzarchitektur für den Aufbau der Software des General Purpose Access Points vorgestellt und genauer beschrieben, bevor das Kapitel mit einer kurzen Zusammenfassung abschließt.

3.1 Klassifizierung heterogener Netzwerkstrukturen

Bei einem typischen heterogenen Netzwerk in einer pervasiven Umgebung, wie es im MuSAMA-Projekt zum Einsatz kommt und in Abbildung 3.1 dargestellt ist, wird ein IP-Router als Zugangspunkt zum Internet genutzt. An diesen Router können stationäre Geräte angeschlossen werden, die wie ein Drucker oder ein eingebettetes System über eine IP-basierte Ethernet-Schnittstelle verfügen. Mobile Geräte wie der PDA, das Notebook oder der Projektor können z.B. über WLAN direkt in das Netzwerk

integriert werden. Weiterhin kommunizieren Bluetooth-basierte Geräte wie der TabletPC und der PDA sowie ZigBee-basierte Sensoren, wie die beiden eingebetteten Systeme jeweils ohne die Nutzung von IP-Adressen direkt miteinander, um Sensordaten zu erfassen, auszutauschen und sie in Form von Kontextinformationen z. B. für eine intelligente Raumsteuerung bereitstellen zu können.

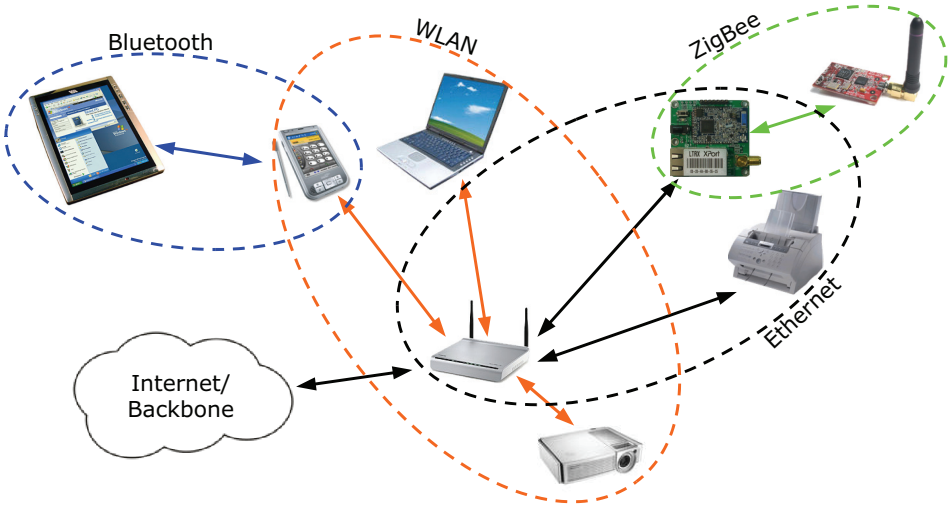

Abbildung 3.1 Kommunikation in einem heterogenen Netzwerk

In diesem Szenario entstehen spezielle *homogene Zellen*, die jeweils aus einer Ansammlung von Geräten gebildet werden, die über Netzwerkschnittstellen derselben Technologie verfügen und sich in physikalischer Kommunikationsreichweite befinden. Je nach Art der Netzwerkschnittstelle können Geräte innerhalb der jeweiligen Zelle entweder drahtgebunden (z. B. per Ethernet) oder drahtlos (z. B. per WLAN, Bluetooth oder ZigBee) miteinander kommunizieren. Besitzt ein Gerät wie z. B. der PDA oder der Router mehrere Schnittstellen unterschiedlicher Technologien, so ist es Teilnehmer mehrerer unterschiedlicher homogener Zellen. Innerhalb einer homogenen Zelle nutzen alle Teilnehmer die gleichen technologiespezifischen Adressierungs- und Kommunikationsmechanismen, die bereits in den Abschnitten 2.2 und 2.3 für aktuelle LAN-, WLAN- und WPAN-Technologien vorgestellt wurden. Da Daten innerhalb einer homogenen Zelle ohne Adress- oder Datenkonvertierungen direkt zwischen den Teilnehmern der Zelle ausgetauscht werden können und Geräte ein spezielles Netzwerk bilden, wird in diesem Zusammenhang auch von einer *horizontalen Netzwerkstruktur* gesprochen. Damit ergeben sich aus dem vorherigen Szenario Zellen bzw. horizontale Netzwerkstrukturen, die in Abbildung 3.2 noch einmal genauer verdeutlicht werden.

In einer Zelle können z. B. ein PDA, ein TabletPC oder andere mit Bluetooth ausgestattete Geräte ein gemeinsames Scatternet aufbauen. In einer anderen Zelle rea-

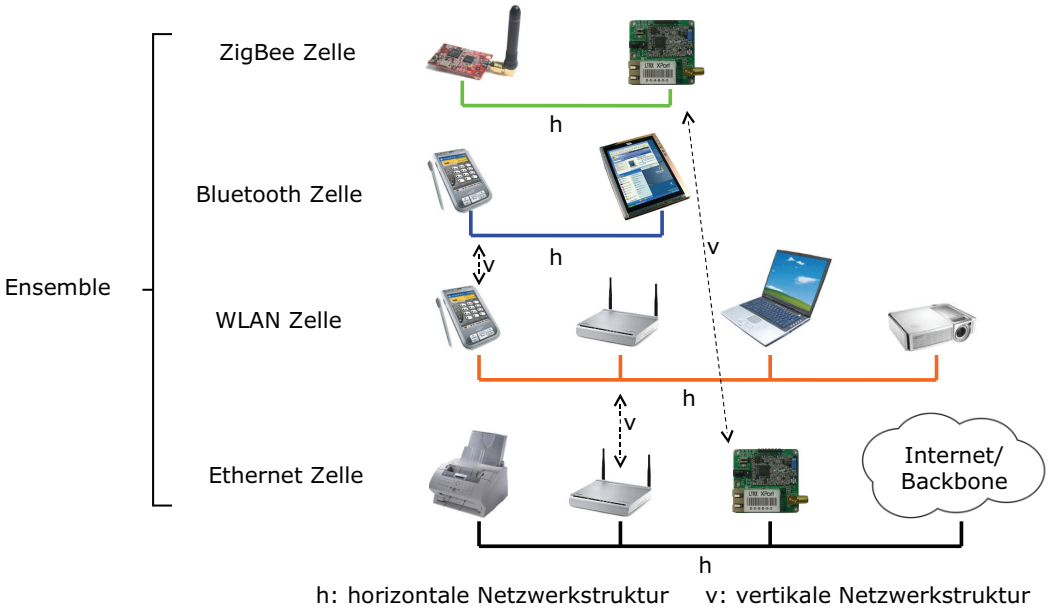

Abbildung 3.2 Von homogenen Zellen zum heterogenen Ensemble

lisieren mit WLAN ausgestattete Geräte ein Ad-hoc-Netzwerk. Diese Zellen haben einen sehr dynamischen Charakter, da Geräte spontan einem drahtlosen Netzwerk beitreten oder es verlassen können. Drahtgebundene Zellen können unter Umständen auch dynamisch sein, da Geräte beliebig ein- bzw. ausgeschaltet und Kabel unvorhersehbar ein- oder ausgesteckt werden können; im Allgemeinen sind sie jedoch eher statisch.

In den Darstellungen 3.1 und 3.2 wird auch deutlich, dass einige Geräte über Schnittstellen in mehreren Zellen verfügen (z. B. der PDA, der Router und das eingebettete System). Diese Geräte sollen, abhängig von ihren Kommunikationstechnologien, als Vermittler zwischen den technologiefremden Zellen eingesetzt werden. Ihre Aufgabe besteht dann darin, zwei horizontale Netzwerkstrukturen durch eine *vertikale Netzwerkstruktur* so zu verbinden, dass die Adressierung und Kommunikation der Geräte aus der einen Zelle mit den Geräten der anderen Zelle ermöglicht wird. Im Allgemeinen sind dabei Mechanismen für technologiespezifische Adress- bzw. Datenkonvertierungen zwischen beiden horizontalen Netzwerkstrukturen vonnöten. Hierbei sei auf den im Abschnitt 2.4 vorgestellten aktuellen Stand der Forschung verwiesen, der bisherige Ansätze zur Interoperabilität unterschiedlicher Netzwerktechnologien auf der Ebene der Internetschicht und der Anwendungsschicht des TCP/IP-Referenzmodells vorgestellt hat. Eine Verknüpfung mehrerer horizontaler Netzwerkstrukturen durch vertikale Netzwerkstrukturen führt dazu, dass die homogenen Zellen wie in Abbildung 3.2 dargestellt zu einem *heterogenen Ensemble* kombiniert werden, das durch

eine allgemeingültige Adressierung und Kommunikation zwischen beliebigen Teilnehmern sämtlicher beteiligter Zellen gekennzeichnet ist und so jedes Gerät transparent mit jedem anderen Gerät im Ensemble kommunizieren kann. Da einige der homogenen Zellen dynamisch sind, besitzt auch ein aus ihnen zusammengesetztes Ensemble einen dynamischen Charakter.

In einer pervasiven Umgebung kommen unterschiedliche Geräte spontan zusammen und nutzen verschiedene Arten an Netzwerkschnittstellen zur Kommunikation. Dabei tritt die Frage auf, ob es mit der Art und der Anzahl der in einer Umgebung vorhandenen Netzwerkschnittstellen zu einem bestimmten Zeitpunkt überhaupt möglich ist, die Schnittstellen in homogene Zellen einzuordnen und sie zu einem heterogenen Ensemble zu kombinieren, indem jedes Gerät mit jedem anderen Gerät kommunizieren kann. Dazu ist zu untersuchen, ob zu diesem Zeitpunkt eine logische Konnektivität des Ensembles vorliegt. Bevor die folgenden Abschnitte näher auf die Bestimmung der logischen Konnektivität eines Ensembles eingehen, soll nun kurz erläutert werden, was in dieser Arbeit unter physikalischer und logischer Konnektivität von Netzwerkschnittstellen verstanden wird.

Die Kommunikation in einer homogenen Zelle erfolgt unter der Voraussetzung, dass Netzwerkschnittstellen als real existierende Komponenten sowohl physisch als auch logisch miteinander verbunden sind. Dabei bedeutet die *physikalische Konnektivität* zweier Schnittstellen, dass sie derselben Netzwerktechnologie angehören und sich in gegenseitiger Kommunikationsreichweite befinden, die z. B. bei der Nutzung von WLAN, Bluetooth oder ZigBee von der Reichweite der Funkübertragung abhängt. Diese Annahme beschränkt die physikalische Konnektivität auf eine technologieabhängige homogene Zelle. Die *logische Konnektivität* zweier Schnittstellen bedeutet dagegen, dass beide Schnittstellen über eindeutige Adressen verfügen und auf der Internet-, der Transport- oder der Anwendungsschicht des TCP/IP-Referenzmodells miteinander Daten austauschen können. Dies trifft nicht nur für die Schnittstellen einer homogenen Zelle zu, sondern ermöglicht es weiterhin homogene Zellen wie eine WLAN- und eine Bluetooth-Zelle durch ein Gerät miteinander zu verbinden, das über Schnittstellen beider Technologien verfügt und somit Teilnehmer beider Zellen ist. Dieses Gerät kann dann z. B. durch Realisierung der Interoperabilität auf der Anwendungsschicht (siehe Service Proxying in Abschnitt 2.4.2) die logische Konnektivität zwischen den Schnittstellen der unterschiedlichen Zellen ermöglichen. Für ein heterogenes Ensemble bedeutet logische Konnektivität also, dass alle homogenen Zellen durch Geräte miteinander verbunden werden, die mit mehreren Netzwerkschnittstellen und der Fähigkeit zur Realisierung der Interoperabilität dieser Zellen ausgestattet sind.

3.2 Bestimmung der logischen Konnektivität von Ensembles

Aufgrund ihrer Allgemeinheit werden Graphen häufig bei der Modellierung realer Vorgänge eingesetzt. Dabei werden Objekte als Knoten und Beziehungen zwischen den Knoten als Kanten bezeichnet, die zusätzliche Bewertungen in Form eines Kantengewichts (z. B. eine Weglänge) tragen können. Eine exakte Beschreibung eines ungerichteten Graphen liefert Definition 3.1.

Definition 3.1 Ungerichteter Graph

Ein ungerichteter Graph $G = (V, E, I)$ ist ein Tripel von Mengen V (der Menge der Knoten), E (der Menge der Kanten) und I (der Inzidenzrelation) mit den Eigenschaften

1) $I \subseteq V \times E$ (ist $(v, e) \in I$ für $v \in V$ und $e \in E$, so heißen v und e inzident)
2) jede Kante $e \in E$ ist zu höchstens zwei Knoten aus V inzident:
$\forall_{e \in E} |(V \times \{e\}) \cap I| \in \{1, 2\}$.

An dieser Stelle sei darauf hingewiesen, dass für die Bestimmung der logischen Konnektivität eines Ensembles in der weiteren Arbeit von ungerichteten schlichten Graphen ausgegangen wird, die nach Definition 3.2 keine Schlingen und parallele Kanten enthalten.

Definition 3.2 Schlichter Graph

Eine Kante $e \in E$ heißt Schlinge gdw.
$|(V \times \{e\}) \cap I| = 1$ (d.h. e ist nur zu einem Knoten inzident).

Zwei Kanten e_1, e_2 heißen parallele Kanten gdw.
$\{v : (v, e_1) \in I\} = \{v : (v, e_2) \in I\}$
(d.h. e_1, e_2 sind zu denselben Knoten inzident).

Ein Graph heißt schlicht gdw.
$G = (V, E)$ enthält keine Schlingen und parallelen Kanten.

Auch die Geräte eines Ensembles können für die Modellierung des Netzwerks als eine Menge von Knoten und die möglichen Verbindungen zwischen ihnen unter Annahme von bidirektionalen Kommunikationsverbindungen als Kanten eines ungerichteten schlichten Graphen dargestellt werden. Die eigentliche Kommunikation obliegt jedoch nicht den Geräten sondern eher den Schnittstellen des Netzes. Daher ist es an dieser Stelle sinnvoller, nicht komplette Geräte, sondern wie in Abbildung 3.3 dargestellt, die unterschiedlichen Netzwerkschnittstellen der Geräte als Knoten zu betrachten [93].

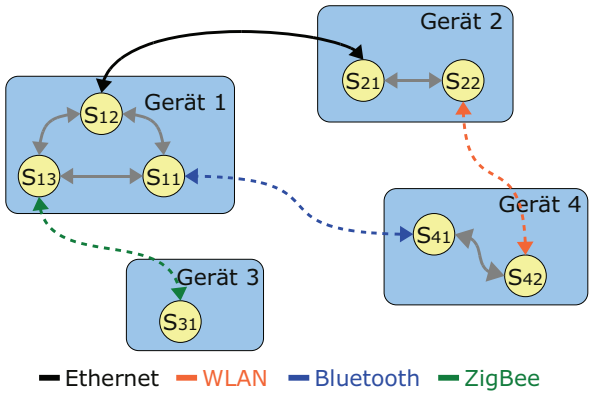

Abbildung 3.3 Beispiel eines heterogenen Netzwerks mit vier Geräten

Durch diese Betrachtungsweise werden die in einem Ensemble vorhandenen Netzwerkschnittstellen als die Menge der Knoten betrachtet, die untereinander abhängig vom Typ der Netzwerkschnittstelle durch spezielle Kommunikationsverbindungen verknüpft sind. Hierbei werden Netzwerkschnittstellen, die durch eine physikalische Konnektivität innerhalb einer Zelle gekennzeichnet sind, miteinander durch Kanten verbunden. Dies ist in Abbildung 3.3 durch die Ethernet-, WLAN-, Bluetooth- und ZigBee-Verbindungen zwischen den Geräten dargestellt. Weiterhin wird angenommen, dass Geräte, die über mehrere Netzwerkschnittstellen verfügen, ihre Schnittstellen nutzen, um zwischen den unterschiedlichen Zellen als Vermittler aufzutreten und eine logische Konnektivität zwischen den Zellen herzustellen. Dadurch sind die Schnittstellen innerhalb eines Gerätes logisch miteinander verbunden (graue Pfeile). Geräte, die diese logische Konnektivität zwischen mehreren Zellen herstellen, werden in der weiteren Arbeit als *Gateway-Knoten* (GK) und die übrigen als *End-Knoten* (EK) eines heterogenen Netzwerks bezeichnet.

Die in Abbildung 3.1 dargestellten Geräte lassen sich somit in Gateway- und End-Knoten eines heterogenen Netzwerks einordnen. Da der PDA, der Router und das eingebettete System über jeweils zwei Schnittstellen verfügen und zum Erreichen eines heterogenen Ensembles zwischen diesen Technologien vermitteln müssen, stellen sie die Gateway-Knoten in diesem Netzwerk dar. Die restlichen Geräte verfügen jedoch nur über eine Schnittstelle und können somit keine Gateway-Funktion übernehmen. Sie stellen daher die End-Knoten des Netzes dar. Im Bereich heterogener Ensembles ergibt sich daraus die Notwendigkeit einer allgemeineren theoretischen Betrachtung der Struktur und der logischen Konnektivität derartiger Netze. Diese Betrachtung der Konnektivität gibt an, inwieweit die Bildung eines heterogenen Ensembles aus einer vorhandenen Menge an Geräten möglich ist und welchen Geräten dabei eine besondere Bedeutung in diesem Netzwerk zukommt. Dazu wird in den folgenden Ab-

schnitten zuerst auf die theoretische Bestimmung der logischen Konnektivität eines heterogenen Ensembles eingegangen, bevor zwei in dieser Arbeit entstandene Verfahren zur Realisierung der logischen Konnektivität innerhalb des Ensembles näher betrachtet werden.

Zur Beschreibung und Bestimmung der logischen Konnektivität eines heterogenen Ensembles wird auf das Szenario aus Abbildung 3.1 zurückgegriffen. Dabei werden die Begriffe *End-Knoten* (EK), *Gateway-Knoten* (GK) und *Schnittstelle* (S) wie bisher beschrieben genutzt. Hinzu kommt die Einführung eines *Schnittstellenvektors* (SV), der sich als ein Spaltenvektor aus den in einem Gateway- oder End-Knoten verfügbaren Netzwerkschnittstellen zusammensetzt. Die Anzahl der Elemente dieses Vektors wird aus den im heterogenen Netz verfügbaren unterschiedlichen Arten an Netzwerkschnittstellen bestimmt. Die Elemente eines SVs werden für einen bestimmten Gateway- oder End-Knoten durch das Vorhandensein oder das Nichtvorhandensein einer Netzwerkschnittstelle in der jeweiligen Technologie mit 1 oder 0 belegt. Dieser Sachverhalt wird im folgenden Beispiel noch einmal verdeutlicht.

Sind in einem Netzwerk Schnittstellen vom Typ Ethernet (E), WLAN (W), Bluetooth (B) und ZigBee (Z) vorhanden, so entstehen Spaltenvektoren mit vier Elementen. Die Belegung des Schnittstellenvektors zu einem bestimmten Knoten ist im jeweiligen Technologieeintrag entweder 1, wenn der Knoten über eine Schnittstelle in der jeweiligen Technologie verfügt, oder 0, wenn der Knoten nicht über eine Schnittstelle in der Technologie verfügt.

$$\text{a) nur Ethernet: } \mathsf{SV} = \begin{pmatrix} \mathsf{E} \\ \mathsf{W} \\ \mathsf{B} \\ \mathsf{Z} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{b) nur ZigBee: } \mathsf{SV} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Durch Aneinanderreihung der SVs, der in einem heterogenen Netzwerk vorhandenen Gateway- und End-Knoten, entsteht eine *Schnittstellenmatrix* (SM) des Netzes, in der ein Spaltenindex ein bestimmtes Gerät und der Zeilenindex eine vorhandene Schnittstelle in diesem Gerät darstellt. Auf diese Weise lassen sich die verschiedenen Schnittstellen mathematisch zusammenfassen, um darauf aufbauend die logische Konnektivität des Ensembles bestimmen zu können. Das folgende Beispiel zeigt die Belegung einer einfachen Schnittstellenmatrix, dessen darunterliegendes Netzwerk aus drei Geräten besteht.

Ein einfaches Beispiel für die Belegung einer Schnittstellenmatrix lässt sich unter Nutzung der Netzwerktechnologien WLAN und Bluetooth durch zwei End-Knoten und einen Gateway-Knoten darstellen. Dabei besitzt der eine End-Knoten eine Bluetooth-

(SV_1) und der andere eine WLAN-Schnittstelle (SV_2). Der Gateway-Knoten verfügt über beide Arten von Schnittstellen (SV_3), was zu der folgenden Schnittstellenmatrix führt.

$$SV_1 = \begin{pmatrix} B \\ W \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ SV_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ und } SV_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ ergibt } SM = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Bei diesem Beispiel wird angenommen, dass sich die Geräte bzw. deren Schnittstellen, die dieselbe Netzwerktechnologie nutzen, in direkter Kommunikationsreichweite befinden. Somit gibt die Definition der Schnittstellenmatrix lediglich eine mögliche logische Konnektivität an. Sie enthält jedoch keine Angaben über deren zeitlich bedingte Existenz, die durch die mögliche Mobilität der Netzwerkteilnehmer beeinflusst wird.

Um die logische Konnektivität eines Ensembels zu überprüfen, ist es zuerst notwendig die Schnittstellenmatrix in eine *Adjazenzmatrix* (AM) nach Definition 3.3 zu überführen [94], die dann einen Graphen der Verbindungen der Schnittstellen innerhalb des Ensembles repräsentiert. Dazu werden die im Ensemble vorhandenen Netzwerkschnittstellen von 1 bis n durchnummeriert und die möglichen Verbindungen zwischen den Schnittstellen in eine $n \times n$-Matrix eingetragen. In ungerichteten schlichten Graphen ohne Mehrfachkanten wird dabei in die i-te Zeile und j-te Spalte eine 1 eingetragen, wenn die i-te und j-te Schnittstelle logisch miteinander verbunden sind. Das heißt, es muss für jedes mit 1 belegte Element in der Schnittstellenmatrix überprüft werden, ob in der jeweiligen Zeile (physikalische Konnektivität mit einer anderen Schnittstelle der gleichen Netzwerktechnologie) oder Spalte (logische Konnektivität mit einer anderen Schnittstelle des gleiches Gerätes) eine weitere 1 steht, die eine Verbindung zwischen den Schnittstellen darstellt. Ist also eine weitere 1 in der Zeile oder Spalte der Schnittstellenmatrix vorhanden, so wird eine 1 in die Adjazenzmatrix eingetragen; anderenfalls eine 0.

Definition 3.3 Adjazenzmatrix eines schlichten Graphen

$G = (V, E)$ sei schlichter endlicher Graph mit $V = \{v_1, ..., v_n\}$, $|V| = n \geqslant 1$. Dann lassen sich die Kanten von E in einer $n \times n$-Matrix beschreiben.

Es sei $a_{ij} = \begin{cases} 1, & \text{falls } v_i v_j \in E \\ 0, & \text{sonst} \end{cases}$

$A = (a_{ij})_{i,j \in \{1,...,n\}}$ heißt die Adjazenzmatrix von G. Für ungerichtete Graphen ist diese Matrix symmetrisch, d. h. $a_{ij} = a_{ji}$ für alle $i, j \in \{1, ..., n\}$.

Mit Hilfe der aus der Schnittstellenmatrix des Ensembles abgeleiteten Adjazenzmatrix kann nun der Zusammenhang bzw. die Verbundenheit des Graphen bestimmt und damit die logische Konnektivität des Ensembles überprüft werden. Dabei können bekannte Algorithmen zur Tiefensuche (Depth-First Search (DFS)) oder Breitensuche (Breadth-First Search (BFS)) [95] eingesetzt werden, mit denen sich überprüfen lässt, ob ein Graph zusammenhängend nach Definition 3.5 ist.

Definition 3.4 Weg in einem Graphen

Ein Weg $W = (v_1, v_2, \ldots, v_p)$ in einem Graphen $G = (V, E)$ ist eine Folge von Knoten gdw. für alle Knoten $v_i, v_{i+1} \in V$ mit $i \in \{1, ..., p-1\}$ existiert eine Kante zwischen v_i und v_{i+1}.

Definition 3.5 Zusammenhängender Graph

Ein Graph $G = (V, E)$ heißt zusammenhängend gdw.
für alle Knoten $u, v \in V, u \neq v$, existiert ein Weg zwischen u und v, mit u als Startknoten und v als Endknoten (vgl. Definition 3.4).

Beide Algorithmen erlauben das vollständige Durchsuchen eines Graphen angefangen von einem Startknoten entlang seiner Kanten, wobei alle Knoten erreicht und für sie eine Reihenfolge des Erreichens definiert werden kann. Dabei kommt es nicht darauf an, dass die Kanten oder Knoten einen Kreis bilden, sondern lediglich, dass alle Knoten erreicht werden. Wenn also alle Knoten der in eine Adjazenzmatrix überführten Schnittstellenmatrix des Ensembles mit Hilfe der Tiefen- oder Breitensuche erreicht werden können, so ist das zugrundeliegende heterogene Ensemble logisch zusammenhängend. In diesem Fall ist die wichtigste Voraussetzung dafür erfüllt, dass jedes Gerät mit jedem anderen Gerät innerhalb des Ensembles kommunizieren kann. In einigen Anwendungen wie z. B. dem Routing in Netzwerken, kommen noch Gewichte für die Kanten hinzu und die Aufgabe besteht dann darin, ein Gerüst mit minimalem Gesamtgewicht oder kürzeste Wege (Routen) zu bestimmen. Dabei können Algorithmen wie z. B. Jarnik, Prim oder Dijkstra angewendet werden [95].

Weitere interessante Parameter eines Schnittstellengraphen sind z. B. die *Knotenzusammenhangszahl*, die die kleinste Anzahl von Knoten angibt, deren Entfernung den Zusammenhang des Graphen zerstört, oder aber auch die *Dichte* des Graphen, die das Verhältnis seiner Kantenzahl zur Kantenzahl eines vollständigen Graphen auf gleich vielen Knoten beschreibt und so ein Maß dafür angibt, wie stark die Knoten im Graphen miteinander verbunden sind (Definition 3.6).

Unter Verwendung der Adjazenzmatrix, der im heterogenen Ensemble vorhandenen Netzwerkschnittstellen, kann weiterhin ermittelt werden, wie wichtig bestimmte Kno-

Definition 3.6 Dichte eines einfachen Graphen

Die Dichte $d\eta(G)$ eines einfachen Graphen $G = (V, E)$ ist das Verhältnis seiner Kantenzahl zur Kantenzahl eines vollständigen Graphen auf gleichvielen Knoten, das heißt: $d\eta(G) = \frac{2|E|}{|V|(|V|-1)}$.

ten (Geräte/Schnittstellen) für das Netzwerk sind. Dabei tauchen in der Literatur häufig die Begriffe *degree centrality*, *betweenness centrality* und *closeness centrality* auf, die angeben, wieviele Kanten einen speziellen Knoten enthalten, wie bedeutend ein Knoten im Graphen ist (da über ihn z.B. zentral geroutet wird) oder wie hoch die Nähe bzw. die Dichte der Knoten im Netzwerk ist. Diese Eigenschaften werden im Bereich der Graphentheorie näher untersucht [96].

Der Ablauf der logischen Konnektivitätsbestimmung eines heterogenen Ensembles kann wie folgt zusammengefasst werden:

1) Bestimmung der Anzahl der Schnittstellentypen im Ensemble

2) Aufstellung der Schnittstellenvektoren der Geräte

3) Aufstellung der Schnittstellenmatrix des Ensembles

4) Überführung der Schnittstellenmatrix in eine Adjazenzmatrix eines Graphen

5) Bestimmung der Konnektivität und weiterer Graph-Parameter

Die beschriebene Konnektivitätsbestimmung eines heterogenen Ensembles wird nun anhand von verschiedenen Beispielen genauer verdeutlicht.

3.3 Konnektivitätsbestimmung an ausgewählten Beispielen

Konnektivitätsbestimmung - Beispiel 1
Die Bestimmung der logischen Konnektivität eines heterogenen Netzwerks wird zuerst am Beispiel eines in Abbildung 3.4 dargestellten minimalen Ensembles erläutert, das aus zwei Zellen besteht. Dabei werden zwei End-Knoten, von denen der eine über eine Bluetooth-Schnittstelle und der andere über eine WLAN-Schnittstelle verfügt, durch einen Gateway-Knoten miteinander verbunden, der beide Schnittstellentypen besitzt.

1) Im Netzwerk befinden sich mit Bluetooth- und WLAN-Schnittstellen genau 2 Schnittstellentypen. Daraus ergeben sich Schnittstellenvektoren der Form:

Abbildung 3.4 Minimales Ensemble mit 4 Netzwerkschnittstellen und zwei Zellen

$$SV = \begin{pmatrix} WLAN \\ Bluetooth \end{pmatrix}$$

2) Die Schnittstellenvektoren für die beteiligten Geräte können dann wie folgt beschrieben werden:

$$EK_1 = \begin{pmatrix} 0 \\ 1_{(S_1)} \end{pmatrix},\ GK_1 = \begin{pmatrix} 1_{(S_2)} \\ 1_{(S_3)} \end{pmatrix} \text{ und } EK_2 = \begin{pmatrix} 1_{(S_4)} \\ 0 \end{pmatrix}$$

3) Aus den Schnittstellenvektoren ergibt sich die folgende Schnittstellenmatrix des Ensembles, in der der Spaltenindex ein bestimmtes Gerät und der Zeilenindex vorhandene Schnittstellentypen darstellt. Diese Schnittstellenmatrix gibt dabei lediglich eine mögliche Kommunikation an, enthält jedoch noch keine Angaben über deren zeitlich bedingte Existenz:

$$SM = \begin{pmatrix} EK_1 & GK_1 & EK_2 \end{pmatrix} = \begin{pmatrix} 0 & 1_{(S_2)} & 1_{(S_4)} \\ 1_{(S_1)} & 1_{(S_3)} & 0 \end{pmatrix}$$

4) Um die Konnektivität dieses Ensembles zu bewerten, wird die Schnittstellenmatrix in eine Adjazenzmatrix AM überführt, die dann einen Graphen der Verbindungen der Schnittstellen innerhalb des Ensembles darstellt. Da bei der Betrachtung der Schnittstellenmatrix von symmetrischen Verbindungen zwischen den Schnittstellen und ungerichteten schlichten Graphen ausgegangen wird, die nach Definition 3.2 keine Schlingen und parallele Kanten enthalten, können die Elemente auf und unterhalb der Diagonalen der Adjazenzmatrix AM auf 0 gesetzt werden. Dabei entsteht eine Adjazenzmatrix AM_r, die auf die relevanten Verbindungen reduziert ist.

$$AM = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & S_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array},\ AM_r = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & S_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

5) Anschließend kann mit Hilfe der Tiefensuche bzw. der Breitensuche bestätigt werden, dass die vorliegende Adjazenzmatrix AM_r und damit auch die darunterliegende Schnittstellenmatrix des Ensembles zusammenhängend ist.

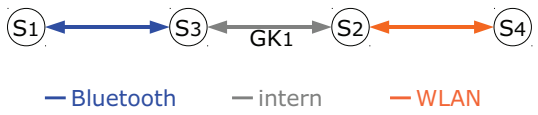

Abbildung 3.5 Grafische Darstellung der Knoten und Kanten der reduzierten Adjazenzmatrix des Ensembles

Die Dichte der in Abbildung 3.5 als Graph dargestellten Adjazenzmatrix, die ein Maß für die Konnektivität (Zusammenhang, Verbundheit) des Ensembles angibt, lässt sich im aktuellen Beispiel nach Definition 3.6 wie folgt berechnen:

$$dn(G) = \frac{2|E|}{|V|(|V|-1)} = \frac{2*3}{4*(4-1)} = \frac{6}{12} = 0,5.$$

Mit einem Wert von 0,5 liegt die Dichte des Graphen schon deutlich unter der maximal möglichen Dichte von 1,0 eines vollständig verbundenen Graphen, indem jeder Knoten direkt mit jedem anderen verbunden ist. Dafür, dass hier mehrere Arten an Netzwerkschnittstellen betrachtet werden, die nur indirekt über einen Gateway-Knoten miteinander kommunizieren können, hat die Dichte aber noch einen recht hohen Wert.

Konnektivitätsbestimmung - Beispiel 2
In diesem Beispiel wird die Konnektivitätsbestimmung an dem in Abbildung 3.6 dargestellten umfangreicheren Ensemble erläutert, das aus 4 homogenen Zellen besteht. Dabei werden 6 End-Knoten durch 3 Gateway-Knoten miteinander verbunden.

1) Im Netzwerk befinden sich mit Ethernet-, WLAN-, Bluetooth- und ZigBee-Schnittstellen genau 4 Schnittstellentypen. Daraus ergeben sich Schnittstellenvektoren der Form:

$$SV = \begin{pmatrix} Ethernet \\ WLAN \\ Bluetooth \\ ZigBee \end{pmatrix}$$

2) Bei den Geräten handelt es sich um 3 Gateway-Knoten (PDA, Router, Eingebettetes System) und 6 End-Knoten (die restlichen Geräte), deren Schnittstellenvektoren wie folgt belegt sind:

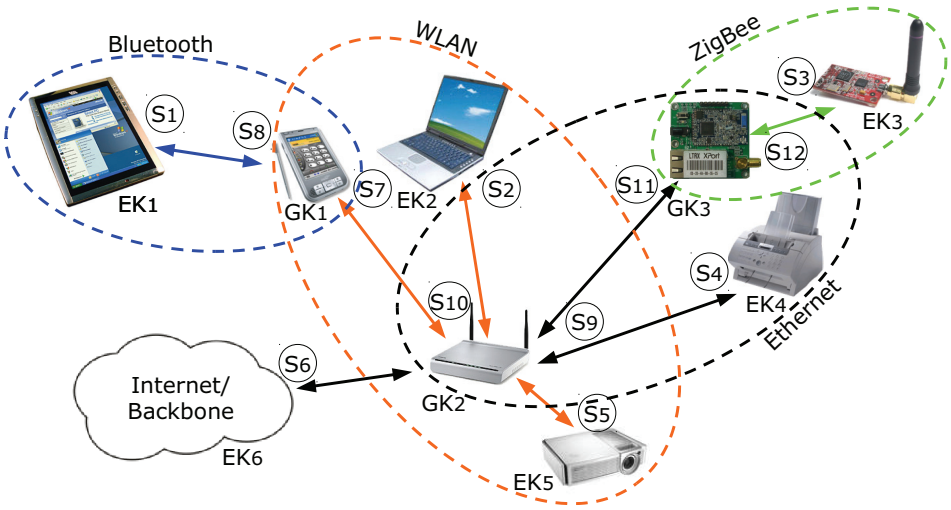

Abbildung 3.6 Ensemble bestehend aus 4 homogenen Zellen

$$
\mathsf{EK}_1 = \begin{pmatrix} 0 \\ 0 \\ 1_{(\mathsf{S}_1)} \\ 0 \end{pmatrix}, \mathsf{EK}_2 = \begin{pmatrix} 0 \\ 1_{(\mathsf{S}_2)} \\ 0 \\ 0 \end{pmatrix}, \mathsf{EK}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1_{(\mathsf{S}_3)} \end{pmatrix},
$$

$$
\mathsf{EK}_4 = \begin{pmatrix} 1_{(\mathsf{S}_4)} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathsf{EK}_5 = \begin{pmatrix} 0 \\ 1_{(\mathsf{S}_5)} \\ 0 \\ 0 \end{pmatrix}, \mathsf{EK}_6 = \begin{pmatrix} 1_{(\mathsf{S}_6)} \\ 0 \\ 0 \\ 0 \end{pmatrix},
$$

$$
\mathsf{GK}_1 = \begin{pmatrix} 0 \\ 1_{(\mathsf{S}_7)} \\ 1_{(\mathsf{S}_8)} \\ 0 \end{pmatrix}, \mathsf{GK}_2 = \begin{pmatrix} 1_{(\mathsf{S}_9)} \\ 1_{(\mathsf{S}_{10})} \\ 0 \\ 0 \end{pmatrix}, \mathsf{GK}_3 = \begin{pmatrix} 1_{(\mathsf{S}_{11})} \\ 0 \\ 0 \\ 1_{(\mathsf{S}_{12})} \end{pmatrix}
$$

3) Aus diesen Schnittstellenvektoren ergibt sich die folgende Schnittstellenmatrix des Ensembles. Wieder stellt der Spaltenindex ein bestimmtes Gerät und der Zeilenindex einen vorhandenen Schnittstellentyp innerhalb des Gerätes dar:

$$
\mathsf{SM} = \begin{pmatrix} \mathsf{EK}_1 & \cdots & \mathsf{EK}_6 & \mathsf{GK}_1 & \cdots & \mathsf{GK}_3 \end{pmatrix}
$$

$$
\mathsf{SM} = \begin{pmatrix}
0 & 0 & 0 & 1_{(\mathsf{S}_4)} & 0 & 1_{(\mathsf{S}_6)} & 0 & 1_{(\mathsf{S}_9)} & 1_{(\mathsf{S}_{11})} \\
0 & 1_{(\mathsf{S}_2)} & 0 & 0 & 1_{(\mathsf{S}_5)} & 0 & 1_{(\mathsf{S}_7)} & 1_{(\mathsf{S}_{10})} & 0 \\
1_{(\mathsf{S}_1)} & 0 & 0 & 0 & 0 & 0 & 1_{(\mathsf{S}_8)} & 0 & 0 \\
0 & 0 & 1_{(\mathsf{S}_3)} & 0 & 0 & 0 & 0 & 0 & 1_{(\mathsf{S}_{12})}
\end{pmatrix}
$$

4) Aus dieser Schnittstellenmatrix können dann die folgende Adjazenzmatrix und die reduzierte Adjazenzmatrix der Größe 12×12 erstellt werden. Zur besseren Reproduzierbarkeit wird die reduzierte Adjazenzmatrix AM_r noch einmal in Abbildung 3.7 als Graph der Verbindungen der Schnittstellen innerhalb des Ensembles dargestellt.

$$\mathsf{AM} = \begin{array}{c} \\ \mathsf{S}_1 \\ \mathsf{S}_2 \\ \mathsf{S}_3 \\ \mathsf{S}_4 \\ \mathsf{S}_5 \\ \mathsf{S}_6 \\ \mathsf{S}_7 \\ \mathsf{S}_8 \\ \mathsf{S}_9 \\ \mathsf{S}_{10} \\ \mathsf{S}_{11} \\ \mathsf{S}_{12} \end{array} \begin{array}{c} \begin{array}{cccccccccccc} \mathsf{S}_1 & \mathsf{S}_2 & \mathsf{S}_3 & \mathsf{S}_4 & \mathsf{S}_5 & \mathsf{S}_6 & \mathsf{S}_7 & \mathsf{S}_8 & \mathsf{S}_9 & \mathsf{S}_{10} & \mathsf{S}_{11} & \mathsf{S}_{12} \end{array} \\ \left(\begin{array}{cccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} \right) \end{array}$$

$$\mathsf{AM}_\mathrm{r} = \begin{array}{c} \\ \mathsf{S}_1 \\ \mathsf{S}_2 \\ \mathsf{S}_3 \\ \mathsf{S}_4 \\ \mathsf{S}_5 \\ \mathsf{S}_6 \\ \mathsf{S}_7 \\ \mathsf{S}_8 \\ \mathsf{S}_9 \\ \mathsf{S}_{10} \\ \mathsf{S}_{11} \\ \mathsf{S}_{12} \end{array} \begin{array}{c} \begin{array}{cccccccccccc} \mathsf{S}_1 & \mathsf{S}_2 & \mathsf{S}_3 & \mathsf{S}_4 & \mathsf{S}_5 & \mathsf{S}_6 & \mathsf{S}_7 & \mathsf{S}_8 & \mathsf{S}_9 & \mathsf{S}_{10} & \mathsf{S}_{11} & \mathsf{S}_{(}12) \end{array} \\ \left(\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) \end{array}$$

5) Durch Anwendung der Tiefensuche bzw. der Breitensuche kann auch für die reduzierte Adjazenzmatrix in diesem Beispiel festgestellt werden, dass der Graph und damit auch die zugrundeliegende Schnittstellenmatrix des Ensembles zusammenhängend ist. Die Anzahl der Knoten beträgt in diesem Beispiel 12 und die Anzahl der Kanten 17. Daraus kann die Dichte des in Abbildung 3.7 dargestellten Graphen wie folgt bestimmt werden:

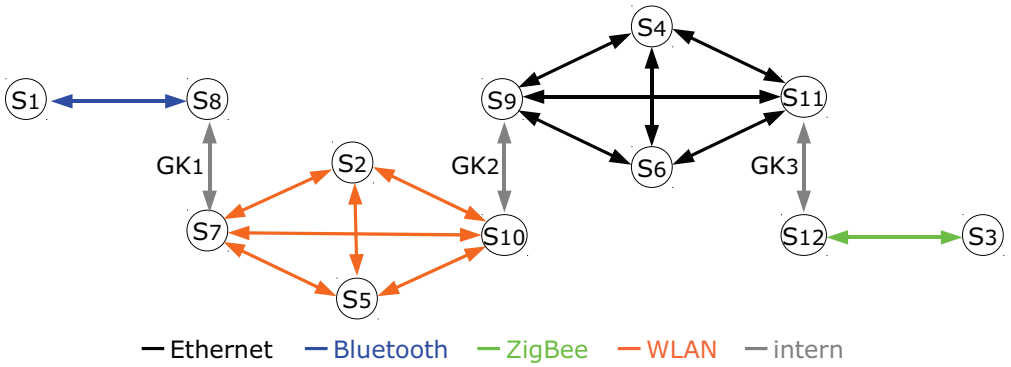

Abbildung 3.7 Grafische Darstellung der Knoten und Kanten der reduzierten Adjazenzmatrix des Ensembles

$$dn(G) = \frac{2|E|}{|V|(|V|-1)} = \frac{2*17}{12*(12-1)} = \frac{34}{132} = 0,2\overline{57}.$$

Die Anwendung der Tiefensuche und die graphische Repräsentation des Graphen in Abbildung 3.7 zeigen, dass der Schnittstellengraph und damit auch das Ensemble logisch zusammenhängend sind. Die Dichte des Graphen hat jedoch mit der Erhöhung der Art der unterschiedlichen Netzwerkschnittstellen deutlich auf einen Wert von $dn(G) = 0,2\overline{57}$ abgenommen.

Konnektivitätsbestimmung - Beispiel 3

Eine weitere Konnektivitätsbestimmung wird am Beispiel des in Abbildung 3.6 dargestellten umfangreicheren Ensembles erläutert. Diesmal wird jedoch ein Gateway-Knoten (der Router) aus dem Ensemble herausgenommen und dessen Schnittstellenvektor (SV8) nicht für die Erzeugung der Adjazenzmatrix verwendet, um die Auswirkung auf den Graphen und besonders dessen Konnektivität kurz darzustellen. Somit ergibt sich eine Adjazenzmatrix der Größe 10 x 10, die in Abbildung 3.8 als Graph dargestellt ist.

Mit 10 Knoten und 10 Kanten ergibt sich in diesem Beispiel eine Dichte des Graphen von nur noch:

$$dn(G) = \frac{2|E|}{|V|(|V|-1)} = \frac{2*10}{10*(10-1)} = \frac{20}{90} = 0,\overline{2}.$$

Diese geringe Dichte des Graphen deutet schon an, dass es mit der geringen Kantenanzahl schwierig ist, einen zusammenhängenden Graphen zu realisieren. Eine Tiefensuche bzw. Breitensuche auf diesem Graphen bestätigt die Andeutung in diesem Beispiel. Eine Kommunikation, in der jedes Gerät mit jedem anderen Gerät Daten austauschen kann, ist hier durch unzureichende logische Konnektivität des heterogenen Ensembles ausgeschlossen.

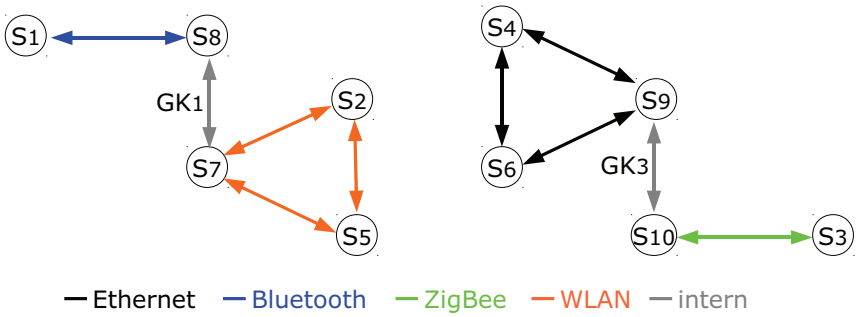

Abbildung 3.8 Grafische Darstellung der Knoten und Kanten der Adjazenzmatrix eines eingeschränkten Ensembles

In der Betrachtung der logischen Konnektivität heterogener Netzwerke und der dabei verwendeten Beispiele kommt den Gateway-Knoten eine besondere Signifikanz für die Konnektivität des Netzes zu. In der vorliegenden Arbeit liegt deshalb der Schwerpunkt der wissenschaftlichen Untersuchung in einem allgemeinen Konzept für einen Gateway-Knoten eines heterogenen Netzwerks. Die End-Knoten sollen möglichst unverändert bleiben, wobei neue Technologien nur durch eine leichte Erweiterung der Gateway-Knoten in das heterogene Ensemble integriert werden können und End-Knoten die für ihre Technologie optimale Adressierungs- (z. B. per MAC-Adressen im Bluetooth- und ZigBee-Netz) und Kommunikationsmethoden (z. B. Dienstnutzung per Bluetooth SDP) beibehalten können.

3.4 Organisationsformen von Gateways für heterogene Netze

Bei der Realisierung eines Gateway-Knotens für heterogene Netze, der homogene Zellen zu einem heterogenen Ensemble kombiniert, können sowohl eine dezentralisierte als auch eine zentralisierte Herangehensweise zur Organisation der Kommunikation innerhalb eines Ensembles betrachtet werden. Beide Herangehensweisen werden in den folgenden Abschnitten kurz mit ihren Vor- und Nachteilen erläutert, bevor auf die in der Implementierung verwendete Herangehensweise genauer eingegangen wird.

3.4.1 Dezentralisierte Gateway-Funktionalität

Die Herangehensweise einer dezentralisierten Gateway-Funktionalität innerhalb eines Ensembles nutzt typischerweise die in Abbildung 3.9 dargestellten und in einer pervasiven Umgebung schon vorhandenen Geräte. Jedes dieser Geräte kann eine geringe

Anzahl unterschiedlicher Netzwerkschnittstellen besitzen und stellt je nach Anzahl dieser Netzwerkschnittstellen und der zusätzlich bereitgestellten Funktionalität entweder einen Gateway-Knoten oder einen End-Knoten dar (vgl. Abschnitt 3.2).

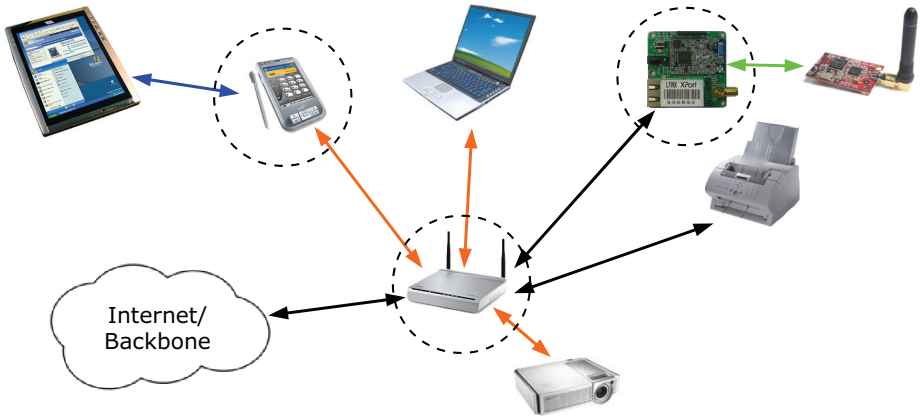

Abbildung 3.9 Typisches Beispiel einer verteilten Gateway-Funktionalität innerhalb eines Ensembles

Auf diese Weise kann ein PDA, der mit einer Bluetooth- und einer WLAN-Schnittstelle ausgestattet ist, einen Gateway-Knoten für die Adressierung und Kommunikation zwischen Bluetooth- und WLAN-Zellen darstellen, wenn er die Funktionalität für eine vertikale Netzwerkstruktur zwischen diesen beiden Zellen bereitstellt. Andere Gateway-Knoten können sowohl durch ein eingebettetes System mit ZigBee- und Ethernet-Schnittstelle als auch durch einen Router mit WLAN- und Ethernet-Schnittstelle realisiert werden. Dabei werden die verschiedenen homogenen Zellen durch mehrere Gateway-Knoten verknüpft um die Struktur eines zusammenhängenden heterogenen Ensembles zu ermöglichen.

Diese auf mehrere Geräte verteilte Gateway-Funktionalität verhindert einen Single Point of Failure und kommt ohne eine fest installierte Infrastruktur aus. Zur Koordination des Ensembles in Form eines verteilten Systems wird jedoch Software benötigt, die über das Kombinieren der homogenen Zellen deutlich hinausgeht und die Adressierung sowie Kommunikation innerhalb eines Ensembles erschwert. Damit z. B. die Kommunikation zwischen dem ZigBee-Sensor und dem Bluetooth-basierten TabletPC stattfinden kann, sind auf dem Kommunikationspfad drei Gateway-Funktionalitäten in Anspruch zu nehmen. Der Kommunikationspfad einer verteilten Herangehensweise wird in Abbildung 3.10 für das Beispiel einer Kommunikation zwischen dem ZigBee-Sensor und dem TabletPC noch einmal genauer dargestellt.

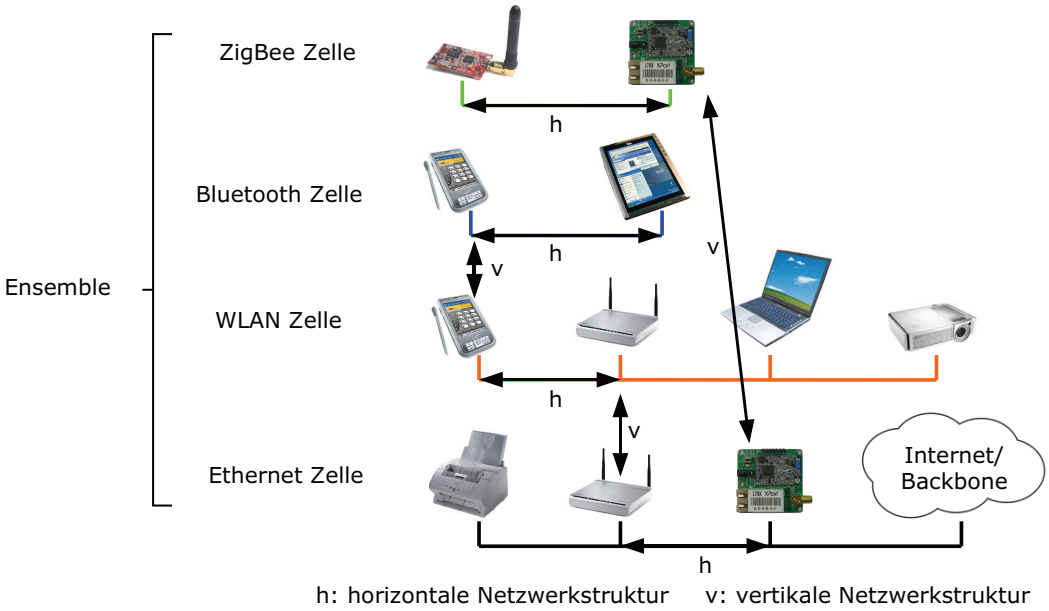

Abbildung 3.10 Beispielhafter Kommunikationspfad für eine dezentralisierte Gateway-Funktionalität

So werden in diesem Szenario ZigBee-Datenpakete vom Sensor an ein eingebettetes System geschickt, in IP-Pakete übersetzt, anschließend per Ethernet und WLAN weitergeleitet und am Ende in Bluetooth-Datenpakete umgewandelt, um sie per Bluetooth an den TabletPC zu versenden. Durch die Inanspruchnahme von drei Gateways auf dem Kommunikationspfad wird nicht nur die Kommunikation erschwert, sondern auch die Latenz erhöht, die die maximale effektive Kommunikationsbandbreite negativ beeinflusst [97].

3.4.2 Zentralisierte Gateway-Funktionalität

Eine zweite Herangehensweise zur heterogenen Kommunikation innerhalb eines Ensembles ist durch die in Abbildung 3.11 dargestellte zentralisierte Gateway-Funktionalität gekennzeichnet.

Die zentralisierte Gateway-Komponente basiert auf einem eingebetteten System, das über mehrere Netzwerkschnittstellen (z. B.Ethernet, WLAN, ZigBee und Bluetooth) verfügt. Jede seiner Schnittstellen repräsentiert dabei einen Teilnehmer einer anderen horizontalen Netzwerkstruktur. Auf diese Weise besitzt er einen Zugangspunkt zu den verschiedenen Zellen und stellt für die Gesamtheit der Zellen einen *General Purpose Access Point* (GPAP) [98][99] dar. Weiterhin verbindet der GPAP die horizontalen Netzwerkstrukturen dieser Zellen durch vertikale Netzwerkstrukturen und ermöglicht

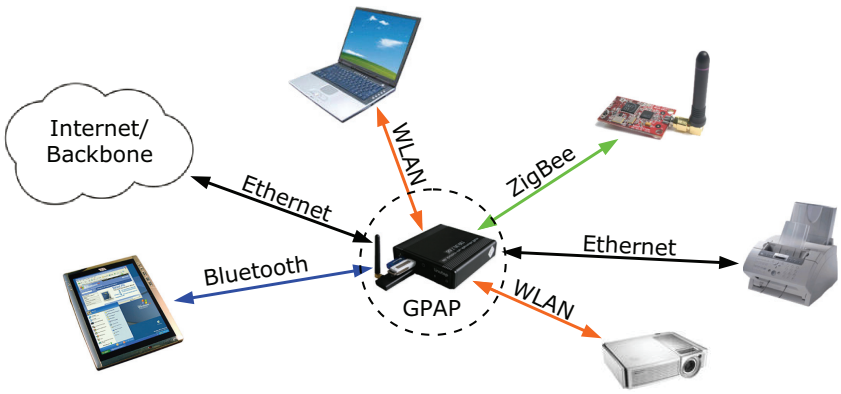

Abbildung 3.11 Zentralisierte Gateway-Funktionalität innerhalb eines Ensembles

so eine Adressierung und Kommunikation zwischen den End-Knoten der verschiedenen Zellen. Damit übernimmt er die Funktion eines Gateway-Knotens zwischen den homogenen Zellen und realisiert ein heterogenes Ensemble. Der GPAP kann dann z. B. Videodaten einer Präsentation per WLAN an den Projektor weiterleiten, Daten mit dem TabletPC per Bluetooth austauschen und Sensorinformationen per ZigBee erhalten. Durch die Konvertierung von Daten zwischen den verschiedenen homogenen Zellen können sie miteinander und mit dem Internet verbunden werden.

Diese Herangehensweise verbessert die Kommunikation zwischen beliebigen Geräten innerhalb eines Ensembles. Angewendet auf das vorherige Beispiel der Kommunikation zwischen einem ZigBee- und einem Bluetooth-basierten Gerät ergibt sich der in Abbildung 3.12 dargestellte und deutlich verbesserte Kommunikationspfad.

Dadurch, dass der GPAP in jeder Zelle über eine Netzwerkschnittstelle verfügt, wird die Anzahl der benötigten Gateway-Knoten auf dem Kommunikationspfad deutlich minimiert. Um Daten vom ZigBee-Sensor zum TabletPC zu übertragen, reicht es nun, sie vom Sensor an den GPAP zu schicken, die ZigBee-Datenpakete in Bluetooth-Datenpakete zu konvertieren und sie vom GPAP aus direkt an den TabletPC zu senden. Auf diese Weise werden die Anzahl der Datenkonvertierungen und gleichzeitig auch die Latenz während der Kommunikation gegenüber einer dezentralisierten Gateway-Funktionalität innerhalb eines Ensembles deutlich verringert. Als kleine Einschränkung dieses Ansatzes sei hier die Menge an verschiedenen Netzwerkschnittstellen zu nennen, die der zentrale GPAP unterstützen muss. Viele gebräuchliche Geräte enthalten nur zwei oder drei verschiedene Netzwerkschnittstellen wie z. B. WLAN, Ethernet und manchmal Bluetooth. Aktuelle eingebettete Systeme (z. B. OpenWrt-Router [100] oder miniaturisierte PCs [101]), die besonders für einen derartigen GPAP geeignet sind, enthalten aber auch leistungsfähige USB-Anschlüsse, über

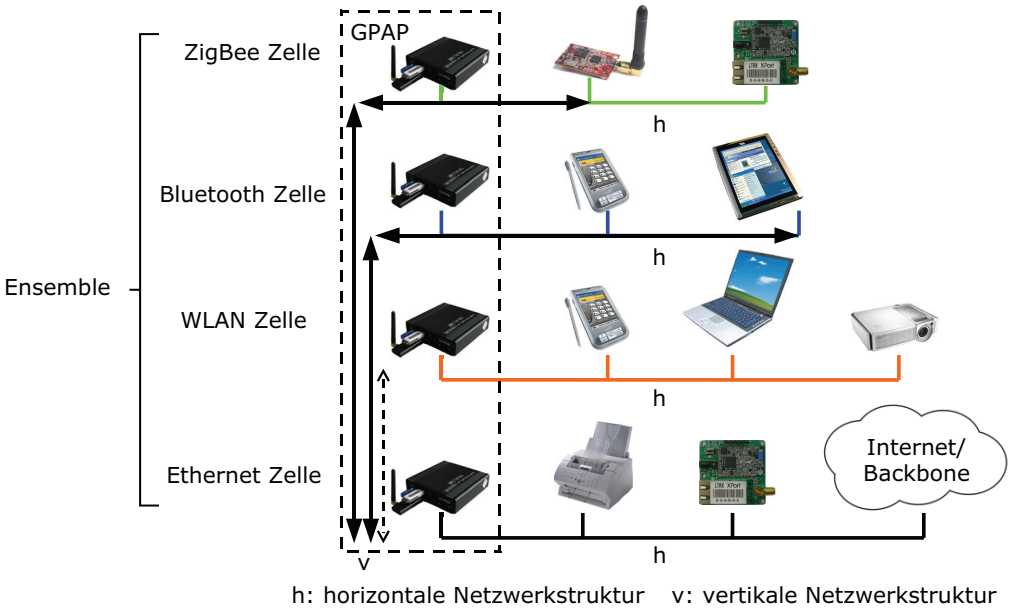

Abbildung 3.12 Beispielhafter Kommunikationspfad für eine zentralisierte Gateway-Funktionalität

die sie ohne großen Aufwand um neue Netzwerkschnittstellen in Form von USB-Sticks erweitert werden können.

3.4.3 Kombination von GPAPs zu einer heterogenen Community

Ein zentralisierter GPAP, der homogene Zellen zu einem heterogenen Ensemble kombiniert, bietet nicht nur die Unterstützung der Kommunikation innerhalb einer räumlich begrenzten heterogenen Umgebung. Auch ein in Abbildung 3.13 dargestelltes Einsatzszenario zur Kopplung räumlich entfernter Umgebungen (z. B. mehrerer Smart Labs, die sich zum Teil in verschiedenen Universitäten befinden) bietet sich an. Hierbei wird die Kommunikation zwischen beliebigen Geräten innerhalb eines Ensembles sowie zwischen unterschiedlichen Ensembles ermöglicht.

In diesem Szenario wird davon ausgegangen, dass sich in jedem Smart Lab ein GPAP befindet, der die Kombination der homogenen Zellen zu einem heterogenen Ensemble innerhalb des Raumes realisiert und so innerhalb dieses Raumes die Adressierung sowie die Kommunikation von jedem Gerät zu jedem anderen ermöglicht. Die GPAPs, die sich jeweils in verschiedenen Räumen befinden, werden durch drahtgebundene (z. B. Ethernet) oder drahtlose (z. B. WLAN, WiMAX) Netzwerkverbindungen miteinander verbunden. Auf diese Weise bilden sie ein eigenes Backbone-Netzwerk für

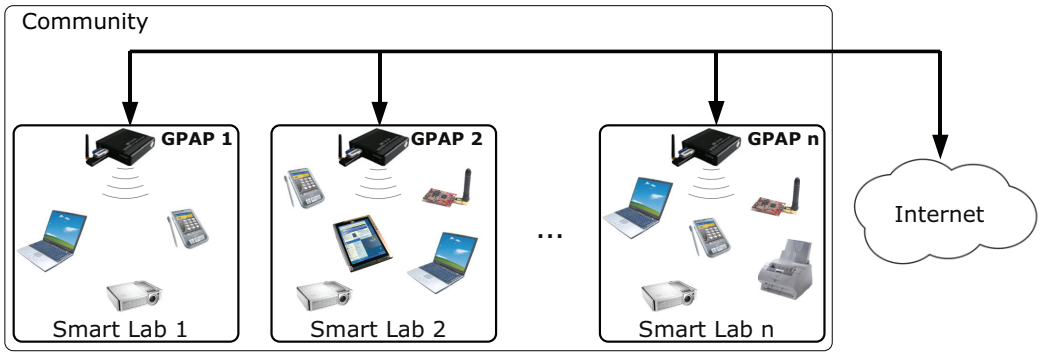

Abbildung 3.13 Kombination von GPAPs und mehreren Ensembles zu einer Community

die verschiedenen Smart Labs. Ein derartiges Backbone wird auch Community genannt [102], in der dann z. B. eine raumübergreifende Service-orientierte Kommunikation zwischen den Geräten unterschiedlicher Smart Labs (oder auch Fachbereiche und Institute einer Hochschule) realisiert werden kann. Erst durch den Einsatz der GPAPs in Form eines Backbones wird eine zentrale Kontrolle über die Kommunikation sowohl innerhalb eines Ensembles als auch zwischen mehreren Ensembles realisierbar. Weiterhin kann für mobile Geräte innerhalb der Community auch ein Roaming von Ensemble zu Ensemble ermöglicht werden, das den pervasiven Umgang mit Geräten im Rahmen des MuSAMA-Projekts unterstreicht.

3.5 Kommunikationsformen in heterogenen Netzen

In den bisherigen Kapiteln lag der Fokus der Betrachtung heterogener Netzwerke besonders auf der Kombination von horizontalen und vertikalen Netzwerkstrukturen zur Erlangung einer logischen Konnektivität innerhalb eines Ensembles (bzw. in einer Community). Dieses Kapitel geht nun genauer darauf ein, wie die logische Konnektivität zwischen den unterschiedlichen Netzwerktechnologien innerhalb eines Gateway-Knotens realisiert werden kann. Dabei stellt dieses Kapitel sowohl die Funktionalität der gegenseitigen Adressierbarkeit von Geräten unterschiedlicher Zellen als auch die Kommunikation zwischen ihnen detaillierter dar.

Beim Blick auf den aktuellen Stand der Forschung auf dem Gebiet der Adressierung und Kommunikation in heterogenen Netzen ergeben sich in der Literatur unterschiedliche Ansätze, die sich auf Grund der hohen Marktdurchdringung besonders auf die Kombination von Bluetooth bzw. ZigBee mit IP-basierten Technologien konzentrieren. Sie wurden schon im Abschnitt 2.4 genauer erläutert und lassen sich, wie in

Abbildung 2.14 dargestellt, in verschiedene Schichten des TCP/IP-Referenzmodells einordnen. In dieser Arbeit dienen sie als Basis für einen eigenen Ansatz zur allgemeingültigen Adressierung sowie Kommunikation in heterogenen Netzen und werden u. a. mit Unterstützung von Mobilität für die End-Knoten und die im heterogenen Netzwerk vorhandenen Dienste erweitert.

Wie im Abschnitt 2.2.4 beschrieben, gelten Service-orientierte Architekturen seit einigen Jahren als mächtiges Konzept zur Lösung von IT-Integrationsproblemen. Die konkreten Funktionen der Geräte werden hierbei in Form von abstrakten Diensten dargestellt um komplexe Einsatzszenarien beherrschen zu können. Somit werden die spezifischen Fähigkeiten einiger Teilnehmer (Provider) einer homogenen Zelle von anderen Teilnehmern (Consumern) dieser Zelle in Form von Diensten genutzt. Das Anbieten, Suchen und Nutzen von Funktionen anderer Geräte innerhalb einer Zelle wird somit deutlich vereinfacht. In heterogenen Netzwerken können innerhalb der verschiedenen Zellen Technologien verwendet werden, die teilweise mit den SOA-Technologien der anderen Zellen inkompatibel sind. Bisher gibt es wie in Abschnitt 2.4.2 beschrieben, nur eingeschränkt nutzbare Ansätze um verschiedene SOA-Technologien zu kombinieren. Diese Ansätze beziehen sich vor allem auf die in Abbildung 2.14 dargestellte Anwendungsschicht. Im Rahmen des GRK MuSAMA wurde mit dieser Arbeit ein allgemeiner Ansatz zur Service-basierten Kommunikation entwickelt, der im Abschnitt 3.5.1 detailliert vorgestellt wird.

Ansätze für eine Adressierung in heterogenen Netzen beziehen sich eher auf die Internetschicht. Um eine echtzeitfähige Kommunikation innerhalb einer Community zu ermöglichen, die zum großen Teil aus mobilen Geräten besteht, werden weiterhin Mechanismen zum heterogenen Handover und vor allem zur allgemeingültigen Adressierung innerhalb des heterogenen Netzwerks benötigt. Das Konzept dieser heterogenen Adressierung wird in Abschnitt 3.5.2 vorgestellt, bevor in Abschnitt 3.6 ein erst durch die Kombination dieser beiden Ansätze mögliches Handover für die Kommunikation mobiler Geräte in heterogenen Netzen beschrieben wird.

3.5.1 Service-basierte Kommunikation in heterogenen Netzen

Aus der Verwendung verschiedener Netzwerktechnologien innerhalb eines heterogenen Ensembles folgt die Nutzung unterschiedlicher, jeweils etablierter SOA-Technologien zur Bereitstellung, Suche und Nutzung von Diensten. So können in IP-basierten Netzwerken SOA-Technologien wie z. B. Web Services, Jini, Bonjour und UPNP eingesetzt werden, die einen eher statischen Charakter besitzen. In Netzwerken, die vor allem durch mobile Geräte mit einer Bluetooth-, oder ZigBee-Schnittstelle gekennzeichnet sind, werden typischerweise IP-freie SOA-Protokolle wie z. B. SDP einge-

setzt, die den Anforderungen einer spontanen und mobilen Dienstnutzung besser gerecht werden. Mit einer immer stärkeren Marktdurchdringung bieten sich gerade Web Services und Bluetooth SDP als beispielhafte Vertreter zweier unterschiedlicher horizontaler Netzwerkstrukturen für ein allgemeingültiges Konzept zur Kombination dieser doch sehr unterschiedlichen Technologien auf der Ebene der Anwendungsschicht des TCP/IP-Referenzmodells an. Daher soll das Konzept eines allgemeingültigen *Service Proxying* am Beispiel des Datenaustausches zwischen diesen beiden SOA-Technologien veranschaulicht werden. Das allgemeine Konzept des Service Proxying wird durch einen Gateway-Knoten realisiert, der zwischen zwei Service-Technologien vermittelt. Dabei werden durch den Gateway-Knoten Dienste der einen Service-Technologie gefunden und in der jeweils anderen Technologie bereitgestellt. Aus Sicht der End-Knoten ist die Suche und Nutzung von Diensten transparent, da technologiefremde Dienste nun in der eigenen Service-Technologie angeboten und genutzt werden können. Damit auch End-Knoten die Dienste der jeweils anderen Technologie nutzen können, ist eine Protokollumsetzung zwischen beiden Service-Technologien auf dem Gateway-Knoten notwendig. Das allgemeine Konzept des Service Proxying wird in Abbildung 3.14 an einem einfachen Szenario verdeutlicht.

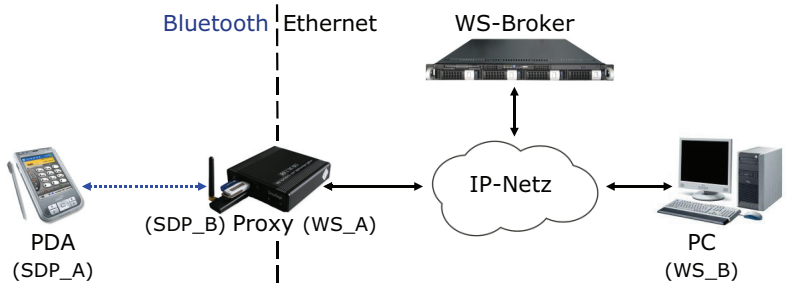

Abbildung 3.14 Darstellung eines rudimentären Service Proxying Szenarios

Dieses einfache heterogene Ensemble besteht aus einer Ethernet- und einer Bluetooth-Zelle, die durch einen Proxy (Gateway-Knoten) verbunden werden. Der PDA und der PC stellen die End-Knoten dieses heterogenen Netzes dar. In der Bluetooth-Zelle sind, wie schon in Abschnitt 2.3.1 beschrieben, SDP sowie OBEX gebräuchliche Standards zur Dienstsuche und zum Datenaustausch zwischen Kommunikationspartnern. In der Ethernet-Zelle bietet das Konzept der Web Services eine anwendungsfreundliche Möglichkeit zum Erstellen, Verwalten und Nutzen von Diensten, wobei zur Kommunikation zwischen Web Services das SOAP-Protokoll verwendet wird.

Die Nutzung eines Bluetooth-Dienstes im IP-basierten Netz kann dabei wie folgt realisiert werden: Zuerst stellt End-Knoten A (PDA) einen Bluetooth-Dienst SDP_A bereit. Der Proxy sucht im Bluetooth-Netz regelmäßig nach Geräten und den von ihnen angebotenen Diensten. Wird der End-Knoten A und dessen Dienst vom Proxy

gefunden, generiert der Proxy einen dem SDP-Dienst entsprechenden Web Service WS_A, der im IP-basierten Netzwerk bereitgestellt und beim WS-Broker registriert wird. Anschließend kann End-Knoten B (PC) Dienste im IP-basierten Netz durch eine Nachfrage beim WS-Broker suchen und den Web Service WS_A finden. Die Nutzung dieses Dienstes wird dann über den Proxy in das Bluetooth-Netz an die Nutzung des Dienstes SDP_A weitergeleitet.

In der anderen Kommunikationsrichtung kann ein Web Service des PCs von einem Bluetooth-basierten PDA wie folgt genutzt werden: Der PC stellt einen Web Service WS_B bereit und registriert ihn beim Broker im IP-basierten Netzwerk. Der Proxy befragt den Broker regelmäßig nach aktuellen Web Services. Sobald der Web Service WS_B gefunden wird, erstellt der Proxy einen äquivalenten und stellvertretenden SDP-Dienst und macht diesen als SDP_B im Bluetooth-Netz verfügbar. Der PDA sucht regelmäßig nach Bluetooth-Geräten in der Nähe und nach deren Diensten. Sobald er den Proxy und dessen Dienst SDP_B gefunden hat, kann er ihn nutzen. Die Dienstnutzung wird dann über den Proxy in das IP-basierte Netz an den Web Service WS_B weitergeleitet.

Da beide Technologien unterschiedliche Transport-Protokolle verwenden, müssen Datenströme auf dem Proxy zwischen OBEX im Bluetooth-Netz und SOAP im IP-Netz umgesetzt werden. Auf diese Weise kann z.B. ein Technologie-übergreifender Nachrichten- oder Dateitransfer zwischen Bluetooth- und IP-basierten Geräten unter Verwendung verschiedener Service-Technologien realisiert werden. Weiterhin wird die Reichweite der Bluetooth-Dienste erhöht, da sie nun auch in weitläufigen kabelgebundenen Netzwerken (IP-Wolke) genutzt werden können.

Mit diesem einfachen Szenario für das Service Proxying zwischen Bluetooth SDP und Web Services werden auch die Rollen der Geräte in beiden Service-Technologien deutlich. Beim Nutzen des Bluetooth-Dienstes SDP_A aus dem IP-Netz heraus, stellt der PDA den Dienstanbieter und der Proxy den Dienstnutzer der Bluetooth Service-Technologie dar (vgl. Abschnitt 2.3.2). Im IP-basierten Netz übernimmt der Proxy die Rolle eines Dienstanbieters, der Broker die Rolle des Dienstverzeichnisses und der PC stellt einen Dienstnutzer dar (vgl. Abschnitt 2.2.4). In der anderen Kommunikationsrichtung tauschen der PDA und der PC die Rollen von Dienstanbieter und Dienstnutzer innerhalb ihrer Technologie. Der Proxy ist dann Dienstnutzer im IP-basierten Netz und Dienstanbieter im Bluetooth-Netz. Damit ergeben sich für die Service-basierte Kommunikation zwischen Bluetooth SDP und Web Services die in Abbildung 3.15 dargestellten Komponenten.

(a) Bluetooth-basierter End-Knoten
ein PDA mit Bluetooth-Schnittstelle

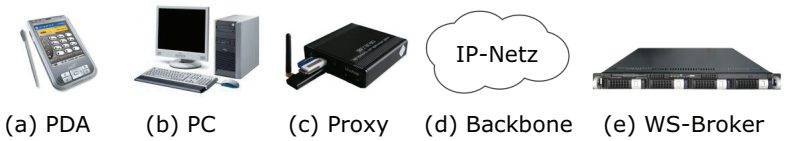

Abbildung 3.15 Komponenten des Service Proxyings zwischen Bluetooth SDP und Web Services

(b) IP-basierter End-Knoten
ein PC mit Ethernet- oder WLAN-Schnittstelle

(c) Gateway-Knoten
ein Service-Proxy mit Bluetooth- und Ethernet- oder WLAN-Schnittstelle

(d) IP-basiertes Backbone
ein IP-basiertes Backbone oder auch das Internet, über das IP-basierte End-Knoten die Gateway-Knoten erreichen können (es wird auch für die Kombination von Ensembles zu einer Community genutzt)

(e) Web Service Broker
ein Dienstverzeichnis, in das Dienstbeschreibungen für z. B. Web Services eingetragen und durch Suchanfragen abgefragt werden können

Aus diesen Komponenten lassen sich neben dem vorherigen Szenario die in Abbildung 3.16 dargestellten und in den folgenden Abschnitten erläuterten Szenarien für eine allgemeine Service-basierte Kommunikation innerhalb einer Community ableiten.

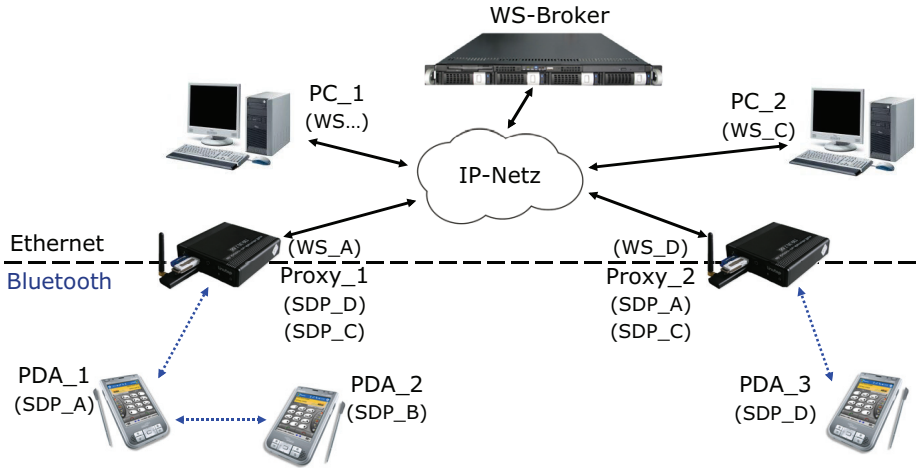

Abbildung 3.16 Ausführliches Szenario des Service Proxyings

Neben der einfachen Service-basierten Datenübertragung innerhalb einer homogenen Zelle, wie sie zwischen einem SDP-Dienstanbieter und einem SDP-Dienstnutzer in einer Bluetooth-Zelle sowie zwischen einem Web Service-Anbieter und einem Web Service-Nutzer in einer Ethernet-Zelle stattfindet, lassen sich zwei weitere Fälle unterscheiden. Der erste Fall betrifft die Kommunikation zwischen einem Bluetooth- und einem IP-basierten End-Knoten. Im zweiten Fall kann durch Service Proxying eine Kommunikation zwischen räumlich entfernten Bluetooth-End-Knoten mit Hilfe mehrerer Proxys als Vermittler realisiert werden. Dabei wird dann zweimal zwischen den Service-Technologien (SDP und Web Services) vermittelt. Da End-Knoten im Allgemeinen durch die Bereitstellung eines entsprechenden Dienstes signalisieren, dass sie in der Lage sind Nachrichten bzw. Dateien entgegen zu nehmen, werden in der folgenden Betrachtung Nachrichten und Dateien stets vom Dienstnutzer ausgehend an den Dienstanbieter übertragen. Die in Abbildung 3.16 dargestellten drei Anwendungsfälle einer Service-basierten Kommunikation werden nun detaillierter erläutert.

Service-basierte Kommunikation innerhalb einer Zelle

Befinden sich zwei Bluetooth-basierte Geräte wie z. B. PDA_1 und PDA_2 in gegenseitiger Funkreichweite (physikalische Verbindung), entsteht eine homogene Bluetooth-Zelle, in der sie eine direkte Service-basierte Datenübertragung durchführen können. Dazu stellt PDA_1 einen Bluetooth Dienst SDP_A bereit, den PDA_2 durch eine Suche nach benachbarten Bluetooth-Geräten und deren Diensten finden und direkt nutzen kann. Der Dienst SDP_D des PDA_3 ist hingegen nicht für ihn nutzbar, da sich die Geräte nicht in räumlicher Funkreichweite und somit in unterschiedlichen Bluetooth-Zellen befinden.

In der IP-basierten Ethernet-Zelle kann PC_2 einen Web Service WS_C anbieten und beim WS-Broker im Dienstverzeichnis registrieren, um anderen Dienstnutzern die WSDL-Beschreibung für diesen Dienst zugänglich zu machen. Dieses Verzeichnis kann dann von anderen Ethernet-basierten Geräten wie dem PC_1 durch eine Suchanfrage beim WS-Broker nach einem bestimmten Diensttyp durchsucht werden. Erhält PC_1 einen Suchtreffer mit Informationen über den Dienstnamen und die Adresse des Anbieters zurück, kann er den Dienst innerhalb seiner Zelle direkt nutzen.

Service-basierte Kommunikation zwischen Zellen

Mit der Kombination zweier homogener Zellen durch einen Service-Proxy wird eine Service-basierte Kommunikation zwischen technologiefremden Zellen erreicht. Dabei wird das zuvor beschriebene Service Proxying zwischen beiden Zellen je nach Ausgangspunkt der Nachrichten-/Dateiübertragung unterschiedlich initiiert und in zwei Richtungen angewandt.

Nutzung eines Web Services aus dem Bluetooth-Netz heraus

Bei der ersten Richtung kann ein Web Service im IP-basierten Netz von einem Gerät in der Bluetooth-Zelle genutzt werden, um Daten in die Ethernet-Zelle zu übertragen. Hierbei ist der Bluetooth-Teilnehmer PDA_1 Konsument eines Dienstes, welcher ursprünglich von dem IP-basierten Gerät PC_2 angeboten wird. Proxy_1 übernimmt die Rolle eines Dienstnutzers im IP-basierten Netz und die eines Dienstanbieters im Bluetooth-Netz um zwischen beiden Service-Technologien zu vermitteln. Der Ablauf dieses Szenarios wird nun mit Hilfe der Abbildung 3.17 detailliert dargestellt.

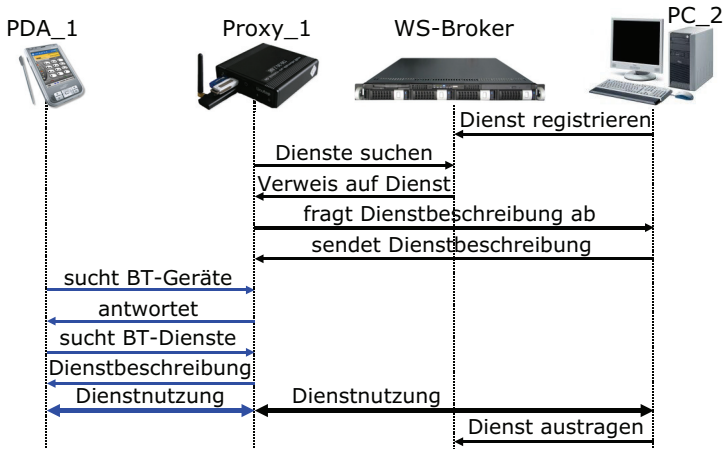

Abbildung 3.17 Sequenzdiagramm für die Nutzung eines Web Services aus dem Bluetooth-Netz heraus

Zuerst registriert PC_2 den von ihm angebotenen Dienst WS_C beim WS-Broker, der ihn in ein lokales Dienstverzeichnis einträgt, damit er später von IP-basierten Nutzern gefunden werden kann. Um diesen Dienst auch in einer Bluetooth-Zelle nutzen zu können, fragen Proxy_1 und Proxy_2 den WS-Broker (speziell sein Dienstverzeichnis) regelmäßig nach aktuellen Diensten ab. Sie bekommen einen Verweis auf den Dienst WS_C zurück und können dann dessen genauere Dienstbeschreibung direkt beim Dienstanbieter abfragen. Mit Hilfe dieser Informationen können sie jeweils einen stellvertretenden Bluetooth-Dienst SDP_C in ihrer lokalen Bluetooth-Zelle bereitstellen. Währenddessen suchen PDA_1, PDA_2 und PDA_3 z. B. alle

30 Sekunden nach in der Nähe befindlichen Bluetooth-Geräten und deren Diensten. Sobald die Proxys den Dienst SDP_C stellvertretend für WS_C anbieten und sich die PDAs in der Reichweite der Proxys befinden, können sie den Dienst finden. Während der Nutzung des Dienstes SDP_C durch PDA_1 wird die Dienstnutzung für den Bluetooth-End-Knoten transparent an den Dienst WS_C des PCs übersetzt. Aus Sicht des Web Service WS_C ist Proxy_1 der Dienstnutzer und aus Sicht des Bluetooth-End-Knotens ist Proxy_1 der Dienstanbieter. Auf diese Weise ist das Service Proxying für die End-Knoten transparent, da sie die eigene Service-Technologie während der Kommunikation nicht verlassen. Sobald der Web Service WS_C beendet wird, trägt er sich im Dienstverzeichnis des WS-Brokers aus. Bei der nächsten Anfrage der Proxys nach diesem Dienst wird festgestellt, dass er nicht mehr vorhanden ist und die Bereitstellung des Dienstes SDP_C wird auch in den Bluetooth-Zellen eingestellt. Bei einer anschließenden durch PDA_1 initiierten Dienstsuche kann der Dienst dann nicht mehr gefunden und genutzt werden.

Nutzung eines SDP-Dienstes aus dem IP-Netz heraus

Bei der entgegengesetzten Richtung kann ein Bluetooth SDP-Dienst von einem Gerät der Ethernet-Zelle genutzt werden, um Daten in die Bluetooth-Zelle zu übertragen. Hierbei ist der IP-Teilnehmer PC_2 Konsument eines Dienstes, welcher ursprünglich von dem Bluetooth-Gerät PDA_1 angeboten wird. Die Rolle von Proxy_1 ändert sich nun in die eines Dienstnutzers im Bluetooth-Netz und die eines Dienstanbieters im IP-basierten Netz, um zwischen beiden Service-Technologien zu vermitteln. Der detaillierte Ablauf dieses Szenarios wird mit Hilfe von Abbildung 3.18 erläutert.

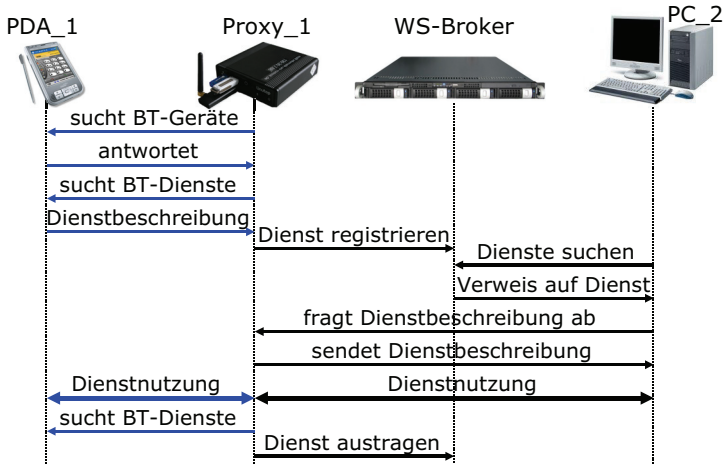

Abbildung 3.18 Sequenzdiagramm für die Nutzung eines SDP-Dienstes aus dem IP-Netz heraus

Zuerst stellt PDA_1 einen Dienst SDP_A bereit, der von anderen Teilnehmern der Bluetooth-Zelle gesucht und genutzt werden kann. Da Proxy_1 ebenfalls Teilneh-

mer dieser Zelle ist und kontinuierlich nach Bluetooth-Geräten in seiner Umgebung und dessen Diensten sucht, kann er ihn ebenfalls finden. Anschließend stellt er stellvertretend für den gefundenen SDP-Dienst SDP_A einen Web Service WS_A im IP-basierten Netz bereit und nimmt dort die Rolle eines Dienstanbieters ein. Der Web Service WS_A wird beim WS-Broker registriert und im Dienstverzeichnis eingetragen. Stellt ein IP-basierter Konsument wie PC_2 eine Suchanfrage zu diesem Dienst an den Broker, bekommt er einen Verweis auf den Dienst, mit dessen Hilfe er die genaue Beschreibung des Dienstes erfragen kann. Diese Informationen können z.B. zum Generieren notwendiger *WS-Schnittstellenklassen* verwendet werden, um die Nutzung des Web Services zu ermöglichen. Die Nutzung des Web Services WS_A durch PC_2 wird wie im vorherigen Szenario wieder über den Proxy an den originalen SDP-Dienst SDP_A in die Bluetooth-Zelle weitervermittelt. Aus Sicht des SDP-Dienstes SDP_A ist Proxy_1 der Dienstnutzer und aus Sicht des IP-basierten End-Knotens PC_2 ist Proxy_1 der Dienstanbieter. Sobald PDA_1 seinen Dienst SDP_A einstellt, kann er vom Proxy_1 nicht mehr gefunden werden und wird von ihm aus dem Dienstverzeichnis des WS-Brokers ausgetragen. Somit steht er auch im IP-basierten Netz nicht mehr zur Verfügung. Auch in diesem Beispiel ist das Service Proxying für die End-Knoten transparent, da sie ihre Service-Technologie während der Kommunikation nicht verlassen.

Service-basierte Kommunikation zwischen entfernten gleichartigen Zellen

Aus der im vorherigen Abschnitt vorgestellten Service-basierten Kommunikation zwischen unterschiedlichen Zellen lässt sich durch Kombination beider Szenarien ein weiteres Szenario realisieren. Dabei werden entfernte Bluetooth-Zellen (Zelle 1: PDA_1, PDA_2 und Zelle 2: PDA_3 in Abbildung 3.16), die zwar dieselbe Service-Technologie nutzen, sich aber nicht in räumlicher Reichweite befinden, durch den Einsatz mehrerer Proxys durch eine als Backbone genutzte IP-basierte Ethernet-Zelle miteinander verbunden. In diesem Szenario bietet PDA_1 einen Dienst SDP_A an, der von dem räumlich entfernten Bluetooth-Gerät PDA_3 genutzt werden soll. Der genaue Ablauf der Service-basierten Kommunikation zwischen diesen entfernten Zellen ist in Abbildung 3.19 dargestellt und wird nun kurz erläutert.

Das Bluetooth-Gerät PDA_1 stellt wieder seinen Dienst SDP_A im lokalen Netz bereit. Dieser Dienst kann von Proxy_1, der in regelmäßigen Abständen nach Bluetooth-Geräten und deren Diensten sucht, gefunden werden. Wie im vorherigen Szenario stellt er stellvertretend für den gefundenen Bluetooth-Dienst einen Web Service WS_A im IP-basierten Netz bereit und registriert diesen beim Broker. Gleichzeitig fragt Proxy_2 den Broker nach aktuellen Diensten ab. Er erhält einen Verweis auf den neu hinzugefügten Dienst WS_A und kann die genaue Beschreibung des

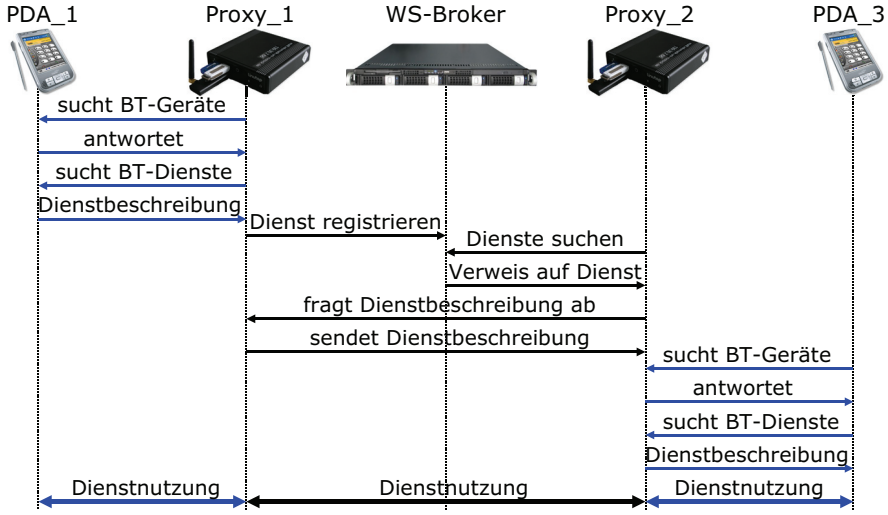

Abbildung 3.19 Sequenzdiagramm für die Nutzung entfernter SDP-Dienste

Dienstes abfragen. Dabei unterscheidet der Proxy z. B. anhand der IP-Adresse in der Beschreibung des Dienstes, ob die Diensteinträge von ihm selber oder einem anderen Teilnehmer erstellt wurden, um Endlosschleifen beim Proxying in beiden Richtungen zu vermeiden (Suche in Technologie 1, Bereitstellung in Technologie 2, neuer Dienst in Technologie 2 gefunden, noch einmal in Technologie 1 bereitstellen, ...). Aus der Beschreibung des fremden Dienstes WS_A generiert Proxy_2 dann einen Bluetooth-Dienst SDP_A, den er in der lokalen Bluetooth-Zelle zur Verfügung stellt. Sucht PDA_3 nun in seiner Umgebung nach Bluetooth-Diensten, kann er den vom Proxy stellvertretend angebotenen Dienst SDP_A finden. Die Nutzung dieses Dienstes wird wieder über Proxy_2 in das IP-basierte Netz und über Proxy_1 in die entfernte Bluetooth-Zelle an den originalen Bluetooth-Dienst SDP_A vermittelt. Sobald PDA_1 seinen Dienst SDP_A einstellt, kann er von Proxy_1 nicht mehr gefunden werden und wird von ihm aus dem Dienstverzeichnis des WS-Brokers ausgetragen. Somit steht er auch im IP-basierten Netz nicht mehr zur Verfügung und kann bei der nächsten Suchanfrage von Proxy_2 an den Broker nicht mehr gefunden werden. Dies führt zum Einstellen des stellvertretenden Bluetooth-Dienstes SDP_A bei Proxy_2. Der Dienst kann dann von PDA_3 nicht mehr weiter genutzt werden.

Aus Sicht der Dienstnutzung in den Bluetooth-basierten Zellen stellt PDA_1 einen Dienstanbieter und Proxy_1 einen Dienstnutzer dar. Ebenso ist Proxy_2 ein Dienstanbieter und PDA_3 ein Dienstnutzer eines Bluetooth-Dienstes. In der IP-basierten Zelle stellt Proxy_1 einen Dienstanbieter und Proxy_2 einen Dienstnutzer eines Web Services dar. Auch in diesem Beispiel ist das Service Proxying für die End-Knoten transparent, da sie ihre Service-Technologie während der Kommunikation

nicht verlassen. Somit lassen sich Dienste einer entfernten Zelle mit diesem Ansatz ohne Kompromisse so nutzen, als ob sie in der eigenen Zelle vorhanden wären.

In diesen Szenarien wird deutlich, dass Proxys (Gateway-Knoten) kontinuierlich in jeder Zelle, für die sie ein Gateway darstellen, nach vorhandenen Diensten suchen, in den jeweils anderen Zellen einen äquivalenten Dienst bereitstellen und die Vermittlung zwischen ihnen solange realisieren, bis der originale Dienst eingestellt wird. Dabei werden Proxys als verteilte Anlaufpunkte bei der Vermittlung zwischen unterschiedlichen Service-Technologien dynamisch zum Dienstnutzer bzw. Dienstanbieter, ohne dass ein zentraler Server zur Steuerung der Kommunikation benötigt wird und End-Knoten die Service-Technologie ihrer Zelle verlassen müssen. Weiterhin kann das beschriebene Service Proxying auf andere Technologien angewendet werden. So können z. B. die Bluetooth-Schnittstellen von PDA_3 und Proxy_2 durch ZigBee-Schnittstellen ersetzt werden und ein heterogenes Service Proxying zwischen Bluetooth SDP und Web Services, sowie zwischen Web Services und den in Abschnitt 2.3.4 beschriebenen ZigBee-Diensten (in Form von Profilen) durch Kombination zur transparenten Kommunikation zwischen Bluetooth- und ZigBee-Geräten genutzt werden.

Bei der Herangehensweise des Service Proxying entstehen Vor- und Nachteile, die an dieser Stelle kurz angesprochen werden sollen. Zu den klaren Vorteilen gehört vor allem die konfliktfreie Adressierung und Kommunikation zwischen den verschiedenen Netzwerktechnologien, da beim Einsatz von SOAs von den direkten Adressierungsarten (MAC-Adressen im Bluetooth-Netz und IP-Adressen im IP-Netz) abstrahiert wird. Weiterhin verlässt ein Dienstnutzer nie seine eigene SOA-Technologie, da ihm beim Service Proxying technologiefremde Dienste in seiner eigenen SOA-Technologie angeboten werden. Diese kann er transparent nutzen, sogar wenn sich wie im letzten Szenario Dienste nicht in räumlicher Reichweite befinden. Dadurch können in einer Netzwerktechnologie die speziell für diese Technologie entworfenen SOA-Verfahren eingesetzt werden und ein ressourcenschonender Datentransfer ermöglicht werden.

Jedoch führt das Service Proxying zu einer hohen Anzahl an Diensten innerhalb eines heterogenen Ensembles, die aber durch Kontextinformationen gefiltert und in ihrer Anzahl begrenzt werden können. Weiterhin müssen auf den Proxys (GPAPs) Datenkonvertierungen durchgeführt werden, z. B. zwischen SOAP (Web Services) und OBEX (SDP). Dabei kann es sich um Plugins für allgemeine/einfache Konvertierungen handeln, die z. B. den Datenteil der Netzwerkpakete von einem in das andere Format kopieren. Es kann sich aber auch um weitaus komplexere Konvertierungen handeln, die z. B. aus einem VoIP-Datenstrom einen Bluetooth-Audio-Strom generieren [103]. Derartige spezielle Konvertierungen führen zu jeweils speziellen Konvertierungsplugins, die dann jeweils einzeln implementiert werden müssen und einen

Mehraufwand für die Software des Proxys bedeuten. Weiterhin wird der Service-basierte Kommunikationsansatz für heterogene Umgebungen von hohen Wartezeiten für die initiale Dienstsuche und -nutzung begleitet, da z. B. die Dienstsuche in Bluetooth-Netzen knapp 10-15 Sekunden in Anspruch nimmt. In IP-basierten Netzen, in denen vorrangig statische Web Services vorhanden sind, werden typischerweise weitaus größere Zeiträume verwendet, um Dienste erneut zu suchen. Im MuSAMA-Umfeld handelt es sich jedoch um eher dynamische Netzwerke, in denen proaktiv erkannt werden soll, wenn ein mobiles Gerät die Reichweite eines Proxys verlässt und in die eines anderen Proxys kommt. Dabei müssen auch die von den Proxys stellvertretend bereitgestellten Dienste der Bluetooth-Geräte neu als Web Services bereitgestellt werden und von den potentiellen Dienstnutzern erneut gesucht und genutzt werden. Die Dauer der Dienstsuche der beteiligten Service-Technologien verhindert hierbei aber derzeit eine unterbrechungsfreie Dienstnutzung. Um Mobilität der End-Knoten und deren Dienste gewährleisten zu können, wird das vorgestellte Konzept des Service Proxying für heterogene Netzwerke in den nächsten Abschnitten um das Konzept einer heterogenen Adressierung erweitert, das durch Maßnahmen in den tieferen Netzwerkschichten für kürzere Handover-Zeiten bei der mobilen Dienstnutzung sorgen soll.

3.5.2 Adressbasierte Kommunikation in heterogenen Netzen

Eine weitere Art der Kommunikation in heterogenen Netzwerken stellt die rein adressbasierte Kommunikation dar, bei der Kommunikationsteilnehmer (End-Knoten) durch Adressen repräsentiert werden. Diese Art der Kommunikation findet in der Netzzugangs- und der Internetschicht des TCP/IP-Referenzmodells statt und ist ebenfalls in Abbildung 2.14 dargestellt. In Computernetzwerken wird dabei auf der untersten Schicht grundsätzlich die MAC-Adresse verwendet, die Netzwerkschnittstellen weltweit eindeutig zugeordnet wird. Sie setzt sich aus der Identifikationsnummer eines beim Institute of Electrical and Electronics Engineers (IEEE) registrierten Herstellers und einer Seriennummer der Schnittstelle zusammen, die direkt vom Hersteller vergeben wird. Mit MAC-48, EUI-48 und EUI-64 (Extended Unique Identifier) [50] stehen 3 Arten an MAC-Adressen zur Verfügung. Die MAC-Adresse wird auch mit den Begriffen Ethernet Hardware Address, Adapter Adresse oder physikalische Adresse bezeichnet und in den MAC-Schichten heute typischer Netzwerktechnologien wie z. B. Ethernet, WLAN, Bluetooth und ZigBee auf der Ebene von Punkt-zu-Punkt-Verbindungen eingesetzt. Bei derartigen Broadcast-Netzwerken identifiziert die MAC-Adresse also jeden Teilnehmer eindeutig und erlaubt es Pakete für bestimmte Empfänger zu markieren. Dies stellt die Basis vieler der auf dem Link

Layer des in Abbildung 2.2 dargestellten OSI-Referenzmodells [21] aufbauenden Protokollschichten dar, die den Aufbau von komplexeren Netzwerken erst ermöglichen.

Für WPANs, die Informationen zwischen festen, portablen oder sich bewegenden Geräten bei geringer Datenrate nur über relativ kurze Distanzen von typischerweise bis zu 10 m austauschen und dabei keine Infrastruktur benötigen, stellt die Adressierung anderer Kommunikationspartner per MAC-Adresse eine einfache und ressourcenschonende Kommunikationslösung dar. Die Nutzung dieser Punkt-zu-Punkt Adressierung wird für Bluetooth im Standard 802.15.1 [104] und für ZigBee im Standard 802.15.4 [68] genauer beschrieben, um eine energieeffiziente und günstige Lösung zur Adressierung und Kommunikation zwischen z. B. mit Bluetooth ausgestatteten Handys oder zwischen Sensoren mit ZigBee-Schnittstelle zu realisieren.

Die Bluetooth-Technologie setzt dabei auf eine 48-Bit-Adresse (MAC-48), die sogenannte BD_ADDR, die ein Gerät eindeutig identifiziert und sich aus 3 Teilen zusammensetzt. In einem 24-Bit-Lower Address Part (LAP) und einem 8-Bit-Upper Address Part (UAP) wird die Hersteller-ID [105] codiert. Darin ist ein Block von 64 zusammenhängenden LAPs für Inquirys reserviert; ein LAP für ein generelles Inquiry und 63 für spezielle Inquirys, die von bestimmten Geräteklassen genutzt werden können. Die restlichen 16 Bit der BD_ADDR können vom Hersteller frei vergeben werden.

In ZigBee-basierten Netzwerken können zur Adressierung 16-Bit- bzw. 64-Bit-Adressen (EUI-64) eingesetzt werden. Gegenüber den gewöhnlichen 48-Bit-Adressen bieten die 64-Bit-Adressen neben einer einfachen Broadcast-Adresse und individueller Geräteadressen, die Möglichkeit der Nutzung von Gruppen- bzw. Multicast-Adressen [50], die genauso wie individuelle Adressen entweder lokal oder universell administriert werden können.

Weiterhin können MAC-48- und EUI-48-Adressen in 64-Bit-Adressen eingekapselt und transportiert werden, um die Migration auf eine global eindeutige 64-Bit-Adresse zu ermöglichen, die auf einem einheitlichen *Organizationally Unique Identifier* (OUI) basiert. Dabei werden Konvertierungen von MAC-48 und EUI-48 nach EUI-64 in beiden Richtungen vorgeschlagen [50] und durch die IEEE versucht, die Verwendung von EUI-64-Adressen gegenüber den 48-Bit-Adressen zu begünstigen.

Auch in WLAN- und Ethernet-basierten Netzen hat die MAC-48-Adresse eine fundamentale Bedeutung für die Adressierung auf dem Layer 2 des ISO-OSI-Referenzmodells (vgl. Abschnitt 2.2). Dabei werden Netzwerksegmente durch Switche so zu lokalen Netzen miteinander verbunden, dass Punkt-zu-Punkt-Verbindungen zwischen den Geräten mehrerer Segmente möglich sind. Um mehrere Teilnetze zu einem größeren Netz zu kombinieren, werden zusätzlich IP-Adressen (IPv4-Adressen (32 Bit

lang) oder IPv6-Adressen (128 Bit lang)) eingesetzt, die auf Basis der Protokolle TCP und UDP eine Ende-zu-Ende-Kommunikation zwischen entfernten Teilnehmern realisieren. Auf diese Weise stellen die IP-Adressen eines beliebigen Datenpakets die Endpunkte und die MAC-Adressen die Zwischenpunkte auf einem Kommunikationspfad im Netzwerk dar. Um den nächsten Zwischenknoten für den Transport eines Pakets zu bestimmen, kann ein Knoten das ARP-Protokoll (vgl. Abschnitt 2.2.3) nutzen und für eine gegebene IP-Zieladresse die Hardwareadresse des nächsten Knotens auf dem Weg zum Ziel bestimmen. Bei IPv6 wird diese Funktionalität durch das *Neighbor Discovery Protocol* (NDP) bereitgestellt. Ein Knoten kann dann anhand der MAC-Adresse des nächsten Knotens ermitteln, ob er sich im selben oder in einem fremden OSI-Layer-2-Netzwerksegment befindet. Befindet sich der Empfänger in einem fremden Segment, sendet er die Datenpakete an einen Router, der die ankommenden Datenpakete nach ihrer Zieladresse analysiert und sie entweder wieder an einen übergeordneten Router (Standard- oder Default-Gateway) oder an ein lokal erreichbares Ziel weiterleitet (routet).

Da eine derartige Adressierung auf IP-Ebene in WPANs wie Bluetooth und ZigBee nicht vorgesehen ist, ist für sie auch ein Routing der Daten in ein entferntes Netzwerk nicht ohne weiteres möglich. In heterogenen Netzen, wie sie im Rahmen des MuSAMA-Projekts eingesetzt werden, ist es jedoch von fundamentaler Bedeutung, dass sich Netzwerkteilnehmer, die unterschiedliche Netzwerktechnologien verwenden, adressieren und miteinander kommunizieren können. Nur so ist es möglich, dass Geräte Informationen über ihren Zustand und ihre Umgebung austauschen und dem Nutzer in seiner Umgebung Assistenz bieten können. Dazu ist jedoch eine Kombination der verschiedenen Adressierungs- und Kommunikationsmethoden der Ethernet-, WLAN- und WPAN-Technologien nötig, für die in den nächsten Abschnitten ein geeigneter Ansatz vorgestellt wird.

Der Ansatz zur Adressierung und Kommunikation zwischen Teilnehmern unterschiedlicher IP- und nicht-IP-basierter Technologien wird, wie in Abbildung 3.20 dargestellt, in diesem Teil der Arbeit am Beispiel von Ethernet und Bluetooth beschrieben. Hierbei werden im Ethernet-basierten Netz IP-basierte Netzwerkverbindungen (Sockets) eingesetzt; im Bluetooth-Netz dagegen MAC-basierte Netzwerkverbindungen. Der generelle Ansatz, der in den folgenden Abschnitten beschriebenen heterogenen Adressierung und Kommunikation, kann für weitere Technologien wie z. B. ZigBee in gleicher Weise angewendet werden.

Eine heterogene adressbasierte Kommunikation wird dabei durch einen Gateway-Knoten ermöglicht, der über Schnittstellen zweier verschiedener Netzwerktechnologien verfügt und zwischen deren unterschiedlichen Adressierungsarten vermittelt. Dabei wird eine transparente Adressierung ermöglicht, die es erlaubt z. B. die IP-basierte

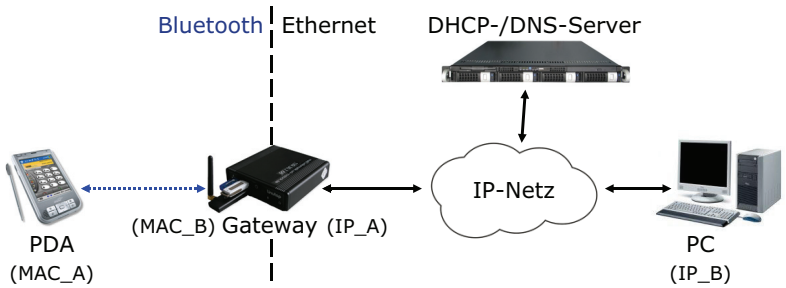

Abbildung 3.20 Einfaches Anwendungsszenario für heterogene Adressierung

und die MAC-basierte Adressierung von Ethernet und Bluetooth so miteinander zu kombinieren, dass MAC-basierte Geräte aus Sicht der IP-basierten Geräte per IP-Adresse adressierbar sind und IP-basierte Geräte aus Sicht der MAC-basierten Geräte per MAC-Adresse (bzw. einer ähnlichen eindeutigen ID) adressiert werden können. Die Verwendung eines entsprechenden Gateway-Knotens innerhalb eines heterogenen Netzwerks macht dann die Nutzung unterschiedlicher Adressierungsarten für die End-Knoten weitestgehend transparent. Ähnlich des in Abschnitt 3.5.1 beschriebenen Service Proxying ist für die Vermittlung der Daten zwischen den verschiedenen Netzwerktechnologien auch hier eine Protokollumsetzung zwischen den je nach Technologie eingesetzten Netzwerkverbindungen (Sockets) auf dem Gateway-Knoten notwendig.

Die Herausforderungen der heterogenen adressbasierten Kommunikation liegen hierbei sowohl in der Erreichbarkeit eines Bluetooth-Gerätes aus dem IP-Netz heraus (PDA vom PC aus adressierbar) als auch in der Erreichbarkeit eines IP-basierten Gerätes aus dem Bluetooth-Netz heraus (PC vom PDA aus erreichbar). Wie im Ansatz der Service-basierten Kommunikation, soll es für die Endgeräte auch hier transparent sein, dass sie mit Geräten einer anderen Technologie kommunizieren. Dabei ist es notwendig, dass IP-basierte Geräte mit anderen scheinbar IP-basierten Geräten per IP-Adresse kommunizieren und nicht-IP-basierte Geräte andere scheinbar ebenfalls nicht-IP-basierte Geräte per MAC-Adresse (bzw. einer ähnlichen ID) ansprechen. Die benötigte Transparenz wird dabei durch ein Gateway bereitgestellt, das die Adressierung und Kommunikation in beiden Richtungen vermittelt und die Heterogenität vor den Endgeräten verbirgt. Damit ergeben sich die in Abbildung 3.21 dargestellten Komponenten.

(a) Bluetooth-basierter End-Knoten
ein PDA mit Bluetooth-Schnittstelle und MAC-Adresse

(b) IP-basierter End-Knoten
ein PC mit Ethernet-Schnittstelle und IP-Adresse

(a) PDA (b) PC (c) Gateway (d) Backbone (e) DHCP-/DNS-Server

Abbildung 3.21 Komponenten für eine heterogene Adressierung zwischen Ethernet- und Bluetooth-Zellen

(c) Gateway-Knoten
ein GPAP mit Bluetooth- und Ethernet-Schnittstelle sowie Gateway-Funktionalität

(d) IP-basiertes Backbone
ein IP-basiertes Backbone oder auch das Internet, über das IP-basierte End-Knoten die Gateway-Knoten erreichen können

(e) DHCP-/DNS-Server
der DHCP-Server dient der Vergabe von IP-Adressen im IP-Netz; der DNS-Server dient der Registrierung der Namen der beteiligten Geräte und deren IP-Adressen

In den folgenden Abschnitten wird zuerst die Herangehensweise der für die End-Knoten transparenten Adressierbarkeit von Bluetooth-Geräten aus dem IP-basierten Netz heraus beschrieben, bevor der Ansatz der Adressierbarkeit von IP-basierten Geräten aus dem Bluetooth-Netz heraus erläutert wird. Anschließend wird eine mögliche Umsetzung von IP-Verbindungen auf Bluetooth-Verbindungen vorgestellt, die es auf eine einfache aber allgemeine Weise erlaubt, IP-Sockets, die mit Hilfe von Ports Daten an ein bestimmtes Programm eines End-Gerätes leiten, auf die im Bluetooth-Netzwerk verwendeten Kanalnummern umzusetzen. Auf diese Weise kann ein einfacher allgemeiner Datenaustausch zwischen heterogenen Netzwerken ermöglicht werden, der den überwiegenden Teil der im MuSAMA-Projekt zwischen den Geräten auszutauschenden Daten und Kontextinformationen ausmacht.

Adressierung von Bluetooth-basierten Geräten aus dem IP-Netz heraus
Bei der transparenten Adressierung eines Bluetooth-basierten Gerätes aus dem IP-Netz heraus ist es wichtig, dass das Bluetooth-Gerät seine MAC-basierte Kommunikation beibehält und somit keinen IP-Stack benötigt. Um aus dem IP-Netz heraus über ein Gateway auf dieses Bluetooth-Gerät zugreifen zu können, ist jedoch eine für das Bluetooth-Gerät stellvertretende IP-Adresse nötig. Naheliegend ist es hier, die IP-Adresse des Gateways zu nutzen und an die in IP-basierten Netzwerken typischerweise verwendete Port Address Translation anzuknüpfen, die im Abschnitt 2.2.2

genauer beschrieben wurde. Dies hätte den Vorteil, dass alle Bluetooth-basierten Geräte über die IP-Adresse des Gateways aus dem IP-Netz heraus adressierbar wären. Jedoch wären die Bluetooth-Geräte hinter dem Gateway (hierbei handelt es sich eigentlich um einen NAT-Router) nur durch die Nutzung eines bestimmten Ports und eines dazugehörigen statischen Eintrags in der NAT-Tabelle des Routers (als Paar aus Port und stellvertretender MAC-Adresse des Bluetooth-Gerätes) erreichbar. Diese Art des Port-Forwarding hat aber wie in rein IP-basierten Netzen den Nachteil, dass die Weiterleitungen statisch beim Router eingetragen werden müssen und Dienste, die auf mehreren Geräten im Bluetooth-Netz auf den gleichen RFCOMM-Kanälen angeboten werden, nur einmal in der NAT-Tabelle vermerkt werden können. So wäre auch deren Adressierung aus dem IP-Netz heraus nur für einen Dienst möglich. Da auf diese Weise weitere Dienste, die im Bluetooth-Netz die gleichen Kanäle nutzen, nicht für das IP-Netz adressierbar sind, stellt ein PAT-basierter Ansatz einen großen Nachteil für heterogene Netzwerke da. Weiterhin würde sich ein PAT-basierter Ansatz auch negativ auf die Unterstützung der Mobilität der End-Geräte auswirken, da mit dem Verlassen der Reichweite eines GPAPs und dem Betreten des Einflussbereiches eines neuen GPAPs, mobile Bluetooth-basierte Geräte nun über die IP-Adresse des neuen GPAPs vom IP-Netz aus erreichbar wären, was unweigerlich zum Abbruch der vorherigen Netzwerkverbindungen führen würde.

Einen eleganteren Ansatz für die Adressierbarkeit von Bluetooth-Geräten aus dem IP-Netz heraus stellt die mit dieser Arbeit entstandene Nutzung sogenannter *virtueller IP-Adressen* für nicht-IP-basierte Geräte dar. Dabei wird wieder davon ausgegangen, dass die im Bluetooth-Netz vorhandenen Geräte nicht mit einem IP-Stack ausgestattet sind und per MAC-Adressen miteinander kommunizieren. Um eine eindeutige Adressierbarkeit der Bluetooth-Geräte aus dem IP-Netz heraus zu ermöglichen, ordnet das Gateway jedem Gerät im lokalen Bluetooth-Netz eine virtuelle IP-Adresse zu, unter der es das Bluetooth-Gerät im IP-Netz erreichbar macht. Dabei wird auf dem im Abschnitt 2.2.2 beschriebenen Basic-NAT-Verfahren aufgebaut. Die Verfahrensweise der Zuordnung einer virtuellen IP-Adresse für Bluetooth-Geräte wird nun anhand des Szenarios aus Abbildung 3.20 und des Sequenzdiagramms in Abbildung 3.22 detaillierter vorgestellt.

Das Gateway (der GPAP) fragt durch regelmäßige Inquirys die im lokalen Bluetooth-Netz vorhandenen Geräte, deren MAC-Adresse und deren Namen ab. Anschließend wird für jedes gefundene Gerät eine IP-Adresse beim DHCP-Server erfragt. Dazu sendet das Gateway zunächst ein DHCPDISCOVER Broadcast-Paket an das IP-Netz (IP Lease Request), das die MAC-Adresse des Bluetooth-Gerätes als Client Identifier enthält. Ist ein DHCP-Server vorhanden, antwortet dieser mit einem DHCPOFFER Paket, in dem ein Vorschlag für eine IP-Adresse enthalten ist (IP Lease Offer). Entspricht das DHCPOFFER den Kriterien des DHCP Clients (in diesem Fall dem Gate-

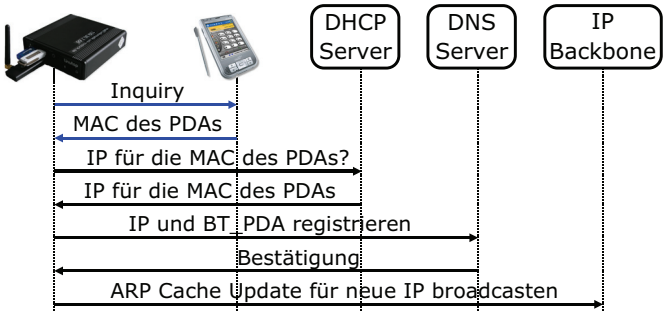

Abbildung 3.22 Vorbereitungsphase zur Nutzung einer virtuellen IP-Adresse

way stellvertretend für das Bluetooth-Gerät), wird das IP Lease beidseitig bestätigt. Benötigt das Gateway mehrere IP-Adressen für lokale Bluetooth-Geräte, so kann die Client ID mit der jeweiligen MAC-Adresse variiert werden. Auf diese Weise kann ein Gateway für jedes lokale Bluetooth-Gerät eine IP-Adresse anfragen und reservieren, die dann als IP-MAC-Paar beim Gateway gespeichert wird. Da die Bluetooth-Geräte zwar über diese IP-Adressen angesprochen werden sollen, sie aber selber nichts davon mitbekommen, werden sie in der weiteren Betrachtung als *virtuelle IP-Adressen* bezeichnet. Anschließend werden die virtuellen IP-Adressen und die dazugehörigen Namen bei einem DNS-Server registriert und das IP-Netz durch einen ARP Cache Update Broadcast über die neue IP-Adresse und ihre Erreichbarkeit über das Gateway informiert.

Nachdem die Vorbereitungsphase mit der Beschaffung der virtuellen IP-Adresse für das Bluetooth-Gerät und der Bekanntmachung der Route zu dieser IP-Adresse abgeschlossen ist, kann eine Kommunikation von einem IP-basierten Gerät zu einem Bluetooth-Gerät, wie im Sequenzdiagramm in Abbildung 3.23 dargestellt, initiiert werden.

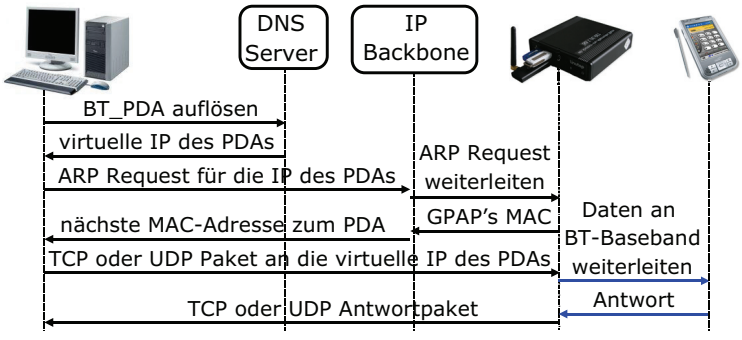

Abbildung 3.23 Nutzung der virtuellen IP-Adresse

Um eine Kommunikation mit dem Bluetooth-Gerät durchführen zu können, muss dem IP-basierten PC entweder die virtuelle IP-Adresse oder der Name des Bluetooth-Gerätes bekannt sein. Angenommen die virtuelle IP-Adresse ist noch unbekannt, so kann der PC den Namen des Bluetooth-Gerätes mit Hilfe des DNS-Servers in dessen virtuelle IP-Adresse auflösen. Anschließend wird in IP-basierten Netzen typischerweise ein ARP-Request mit der IP-Adresse des Kommunikationspartners in Form eines Broadcasts an das lokale IP-Netz gemacht, um die Zuordnung zwischen der Netzwerkadresse der Internetschicht und der physikalischen Hardwareadresse aufzulösen. Dadurch kann in Erfahrung gebracht werden, ob ein Kommunikationspartner im selben Subnetz vorhanden ist und dessen MAC-Adresse als Destination-Adresse im MAC-Header der Datenpakete eingetragen werden muss, oder ob sich der Kommunikationspartner in einem fremden Subnetz befindet, das über einen Router oder ein Gateway erreichbar ist und dessen MAC-Adresse im MAC-Header der Datenpakete genutzt wird. Im Beispiel in Abbildung 3.23 befindet sich das Bluetooth-Gerät in einem fremden Subnetz und seine virtuelle IP-Adresse ist über das Gateway erreichbar. Das Gateway übernimmt hier die Rolle des ARP-Responders, empfängt den ARP-Request des PCs nach der virtuellen IP-Adresse des PDAs und erkennt durch das zuvor gespeicherte Paar aus MAC- und virtueller IP-Adresse, dass das Gerät über ihn erreichbar ist. Daraufhin antwortet das Gateway dem PC mit einem ARP-Reply und seiner eigenen MAC-Adresse. Dadurch wird jegliche Kommunikation an den PDA über das Gateway umgeleitet. Anschließend kann dann z. B. eine UDP-Verbindung zu der virtuellen IP-Adresse aufgebaut werden, die eine Bluetooth-Verbindung zwischen dem Gateway und dem Bluetooth-Gerät nach sich zieht. Wie bei der Service-basierten Kommunikation in Abschnitt 3.5.1 sind auch hier wieder entsprechende Transformationen auf dem Gateway nötig, die für eine Kommunikation zwischen unterschiedlichen Netzwerktechnologien ausgeführt werden müssen. Hierbei handelt es sich jedoch um allgemeinere Transformationen zwischen z. B. UDP/TCP- und Bluetooth-Verbindungen, anstatt um Transformationen zwischen höheren Protokollen wie SOAP (Web Services) und OBEX (Bluetooth SDP).

Adressierung von IP-basierten Geräten aus dem Bluetooth-Netz heraus Auch in der anderen Richtung der Initialisierung einer Kommunikation aus dem nicht-IP-basierten Netz heraus in ein IP-basiertes Netz hinein, soll die Adressierung beliebiger End-Knoten möglichst transparent erfolgen. Das bedeutet, dass wie im vorherigen Abschnitt beschrieben, IP-basierte Geräte über virtuelle IP-Adressen mit nicht-IP-basierten Geräten kommunizieren und nicht-IP-basierte Geräte entfernte IP-basierte Geräte per MAC-Adresse adressieren und mit ihnen Daten austauschen sollten.

Hierbei ist es naheliegend, die MAC-Adresse des entfernten IP-basierten Gerätes zu nutzen und Daten an dieses Gerät über ein Default-Gateway zu transportieren. Am

Beispiel der Adressierung und Kommunikation zwischen Bluetooth- und Ethernet-basierten Geräten zeigt sich aber, dass dieser Ansatz nicht ohne weiteres zu realisieren ist. Angenommen der PDA weiß die MAC-Adresse des entfernten IP-basierten Gerätes (des PCs) und schickt Datenpakete an sein Gateway, die diese Adresse enthalten. Da das Gateway nur Verbindungen auf Basis von IP-Adressen mit dem PC realisieren kann, muss es die zu der MAC-Adresse gehörende IP-Adresse wissen, um die Daten richtig weiterleiten zu können. An dieser Stelle müsste das Gateway eine Liste mit MAC-IP-Paaren für das IP-Netz besitzen (was für größere Netze allein schon eher unwahrscheinlich ist). Um eine derartige Liste zu generieren, könnte aber z. B. über die Service-Schicht des GPAPs auf die Liste der in der Community verfügbaren Dienstbeschreibungen zugegriffen werden, um an die relevanten IP-Adressen zu gelangen. Anschließend müssten für diese IP-Adressen die MAC-Adressen ermittelt werden. Befinden sich das Gateway und einer der PCs im selben Subnetz, ist es für das Gateway noch recht einfach, die MAC-Adresse des PCs mit Hilfe seiner IP-Adresse festzustellen. Dazu reicht eine ARP-Anfrage im lokalen Netzwerk aus. Befinden sie sich jedoch in unterschiedlichen Subnetzen, die durch mindestens einen Router miteinander verbunden sind, ist es für das Gateway unter Nutzung von ARP unmöglich die MAC-Adresse des PCs zu bestimmen, da eine ARP-Anfrage mit der IP des PCs nur die MAC-Adresse des nächsten Routers auf dem Weg zum PC liefert, aber nicht die des PCs selbst.

Ein anderes Hindernis dieses Ansatzes liegt in der Bluetooth-Technologie selbst begründet. Da Bluetooth als eine Punkt-zu-Punkt-Technologie entworfen wurde und Kommunikationspartner in einem Piconet direkt angesprochen werden, ist die Funktion eines Default-Gateways, wie sie in IP-basierten Netzwerken verwendet wird, im Bluetooth-Standard nicht vorgesehen. Die Kommunikation innerhalb eines Piconets wird zwar von einem Master bestimmt und jegliche Daten werden über den Master transportiert; die Kommunikation zwischen 2 Clients kann aber mit im Handel erhältlichen Bluetooth-Netzwerkschnittstellen trotzdem nicht eingesehen werden, da diese Daten aus Sicherheitsgründen gar nicht erst an die höheren Schichten, wie den Treiber oder den Bluetooth-Softwarestack des Masters weitergereicht werden. Daher ist es auch in der Praxis nicht möglich, Datenpakete des PDAs an die MAC-Adresse des IP-Gerätes zu adressieren, sie mit dem Bluetooth-Master auf dem Gateway abzufangen und an die IP des PCs weiterzuleiten, da sie beim Bluetooth-Softwarestack des Gateways einfach nicht ankommen.

Aus diesen Gründen müssen Daten vom PDA an ein entferntes IP-basiertes Gerät direkt an die MAC-Adresse des Gateways geschickt werden und die Information, an welches Gerät sie weitergeleitet werden sollen, in der Nachricht selbst in Form einer MAC, IP oder einer eindeutigen ID enthalten sein, damit das Gateway diese ID auf eine reale IP-Adresse auflösen und Daten entsprechend weiterleiten kann.

Auch die Funktion der Auswahl und Nutzung eines Default-Gateways muss auf dem Bluetooth-Gerät selbst nachgebildet werden, damit Daten aus dem Bluetooth-Netz heraus in andere Netze transportiert werden können. Hierbei gibt es die Möglichkeit, das Gateway ähnlich wie in IP-basierten Netzen auf dem Bluetooth-Gerät statisch festzulegen. Für mobile Geräte in heterogenen Umgebungen bietet es sich jedoch an, die in Reichweite befindlichen Gateways regelmäßig zu suchen und bei Vorhandensein mehrerer ein Gateway z. B. abhängig vom RSSI-Wert auszuwählen. Dafür ist es z. B. möglich, dass Gateways ihre Funktion in Form von SDP-Diensten signalisieren und im lokalen Bluetooth-Netz bereitstellen, die wie in Abbildung 3.24 dargestellt von den mobilen Geräten regelmäßig gesucht werden können.

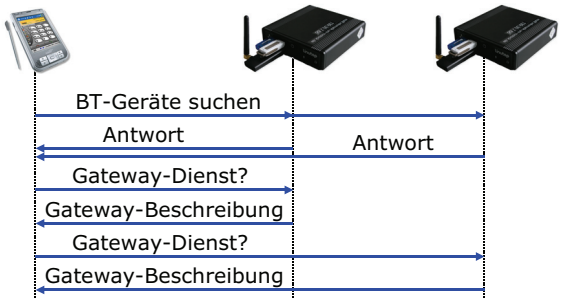

Abbildung 3.24 Auswahl eines geeigneten BT-IP-Gateways

Diese dynamische Gateway-Auswahl stellt eine fundamentale Voraussetzung für ein späteres Soft-Handover der Verbindungen für mobile Geräte dar, bei dem aber noch Absprachen mit und zwischen den Gateways nötig sind, um die Bereitstellung der virtuellen IP-Adresse des Bluetooth-Gerätes durch das neue Gateway zu initiieren und somit die IP-basierte Kommunikation über das neue Gateway dynamisch umleiten zu können.

Für die Nutzung der Gateway-Funktionalität und die Kommunikation mit einem IP-basierten Gerät aus dem Bluetooth-Netz heraus wird im Folgenden angenommen, dass die in Abbildung 3.22 dargestellte und bereits beschriebene Vorbereitungsphase der Zuordnung einer virtuellen IP-Adresse für das Bluetooth-Gerät im IP-Netz abgeschlossen ist. Die Nutzung der Gateway-Funktionalität wird nun mit Hilfe der Abbildung 3.25 genauer erläutert.

Wie in IP-basierten Netzen ist es auch in nicht-IP-basierten Netzen für den Nutzer sehr praktisch, den Namen eines Kommunikationspartners anstatt seiner ID (MAC, IP,...) kennen zu müssen. Darum wird auch hier davon ausgegangen, dass der Name des PCs für den PDA bekannt und dessen ID zunächst unbekannt ist, jedoch durch das Gateway aufgelöst werden kann. Mit der Gateway-Funktionalität sollte

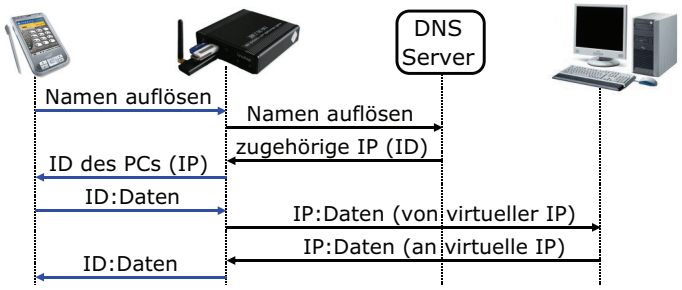

Abbildung 3.25 Nutzung des BT-IP-Gateways

ein GPAP somit auch die Auflösung zwischen einem Namen und einer ID (hier speziell einer IP-Adresse) unterstützen. Da der PC im IP-basierten Netz typischerweise beim DNS-Server mit seinem Namen und seiner IP-Adresse registriert ist, kann der lokale DNS-Server für die Namensauflösung genutzt werden. Die vom DNS-Server zurückgelieferte IP-Adresse wird dann als zum angefragten Namen zugehörige ID an das Bluetooth-Gerät zurückgeliefert. Handelt es sich dabei um einen PC im selben Subnetz, könnte die ID z. B. auch eine eindeutige MAC-Adresse sein. Die Zuordnung aus ID, Name und IP wird auf dem Gateway in Form einer Liste gespeichert. Anschließend kann das Bluetooth-Gerät die an den PC zu übertragenden Daten an die MAC-Adresse des Gateways schicken, wobei die ID des PCs im Paket enthalten sein muss. Das Gateway nutzt diese ID um die IP-Adresse des PCs aus der zuvor gespeicherten Liste auszulesen und die Daten an den PC weiterzuleiten. Dabei ist es wichtig, dass für die IP-basierte Kommunikation nicht die IP-Adresse des Gateways sondern die virtuelle IP-Adresse des PDAs als Absendeadresse genutzt wird. Auf diese Weise können die Antwortpakete automatisch, wie in den vorherigen Abschnitten beschrieben, an die virtuelle IP-Adresse des PDAs zurückgeschickt und durch das Gateway an den PDA übertragen werden.

Nachdem die Ansätze des Service Proxying und der heterogenen Adressierung in den letzten Abschnitten vorgestellt wurden, wird im folgenden Abschnitt genauer auf die Unterstützung der Mobilität der Geräte und ihrer Dienste eingegangen. Dabei wird die Kombination der Service-basierten Kommunikation mit der heterogenen Adressierung detaillierter vorgestellt, die eine Dienst-Mobilität beim Ortswechsel zwischen den Ensembles innerhalb einer Community ermöglichen soll.

3.6 Unterstützung der Mobilität in heterogenen Netzen

Am Anfang dieses Abschnitts wird zunächst ein Szenario vorgestellt, das die gewünschte Mobilität eines Gerätes bzw. seiner Dienste bei Standortwechsel innerhalb

einer Community darstellt. Anschließend wird auf die dadurch bedingten Auswirkungen für den Ansatz der heterogenen Adressierung sowie das Service Proxying eingegangen.

In dem in Abbildung 3.26 dargestellten Szenario werden drei GPAPs über ein IP-Netz zu einem Backbone einer Community verbunden (vgl. Abschnitt 3.4.3). Sie besitzen eine Ethernet- und eine Bluetooth-Schnittstelle und ermöglichen jeweils den Aufbau eines Ensembles, in dem sie durch Service Proxying die Ethernet- und die Bluetooth-Zelle miteinander verbinden und Bluetooth-basierte Geräte über eine virtuelle IP-Adresse im IP-Netz erreichbar machen. Dabei wird davon ausgegangen, dass sich die Einflussbereiche ihrer Bluetooth-Schnittstellen etwas überschneiden. Das führt dazu, dass sich ein mit Bluetooth ausgestatteter mobiler PDA beim Bewegen durch die Community jederzeit in der Bluetooth-Reichweite von ein bis zwei GPAPs befindet.

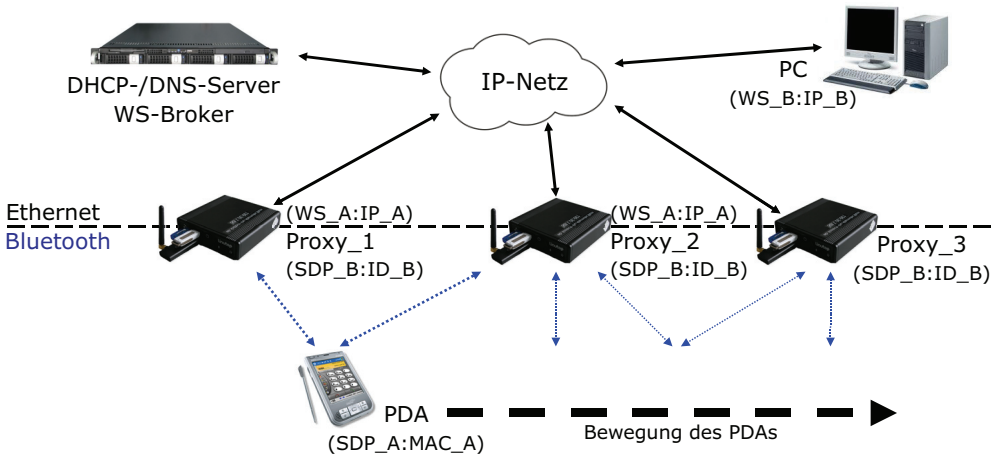

Abbildung 3.26 Szenario für heterogene Adressierung und Mobilität

Im dargestellten Szenario bewegt sich der PDA mit seinem SDP-Dienst SDP_A von der Bluetooth-Zelle des Proxy_1, über die Bluetooth-Zelle von Proxy_2 in die Zelle von Proxy_3 hinein. Als erstes findet Proxy_1 den PDA und stellt dessen SDP-Dienst SDP_A als Web Service WS_A im IP-basierten Netz bereit. Weiterhin macht Proxy_1 den PDA über eine virtuelle IP-Adresse (IP_A) im IP-Netz erreichbar. Sobald der PDA in die Reichweite von Proxy_2 kommt, muss auch dieser den Dienst SDP_A als Web Service WS_A im IP-Netz bereitstellen. In der Übergangsphase ist der Dienst des PDAs also über zwei Proxys im IP-Netz verfügbar. Kurze Zeit später verlässt der PDA die Reichweite von Proxy_1, der daraufhin den Dienst WS_A im IP-Netz wieder einstellt. Der Dienst SDP_A ist nun nur noch über Proxy_2 als WS_A verfügbar. IP-basierte Geräte können diesen Dienst per Service Proxying erreichen. Problematisch ist es jedoch für IP-basierte Geräte, die

zuvor den stellvertretenden Dienst WS_A von Proxy_1 genutzt haben und dessen Dienstnutzung nun über den Proxy_2 umgeleitet werden muss.

An dieser Stelle ist ein unterbrechungsfreier Wechsel der Nutzung des Dienstes WS_A von Proxy_1 zu Proxy_2 wünschenswert, der einen nahtlosen und für den Nutzer transparenten Übergang der Kommunikation ermöglicht. Da im Szenario davon ausgegangen wird, dass sich die Bluetooth-Zellen der Proxys teilweise überschneiden und sich die mobilen Geräte von Bluetooth-Zelle zu Bluetooth-Zelle bewegen, bietet sich hier ein vertikales Soft Handover [106] an, bei dem die aktuelle Verbindung eines mobilen Gerätes zu einem Proxy erst dann getrennt wird, wenn eine neue Verbindung zu einem benachbarten Proxy etabliert wurde. Bevor eine Verbindung mit der Bekanntmachung der virtuellen IP-Adresse IP_A durch Proxy_2 (mit Hilfe eines ARP Cache Updates im IP-Netz) über ihn umgeleitet werden kann, ist aber noch eine Handover-Entscheidung [107] und eine Absprache zwischen den Proxys notwendig, auf die im folgenden Abschnitt auf der Ebene der heterogenen Adressierung genauer eingegangen wird.

3.6.1 Mobilität auf der Ebene der heterogenen Adressierung

Mit der Mobilität der End-Knoten kommen auf die Ebene der heterogenen Adressierung die Aufgabe der Handover-Entscheidung und ein für das Service Proxying transparenter Wechsel der aktuellen Verbindungen (Sockets) zu. Die Vorgehensweise des Handovers von Proxy_1 zu Proxy_2 und die Auswirkungen auf die benötigte Funktionalität innerhalb der heterogenen Adressierung werden nun anhand des Sequenzdiagramms 3.27 genauer erläutert.

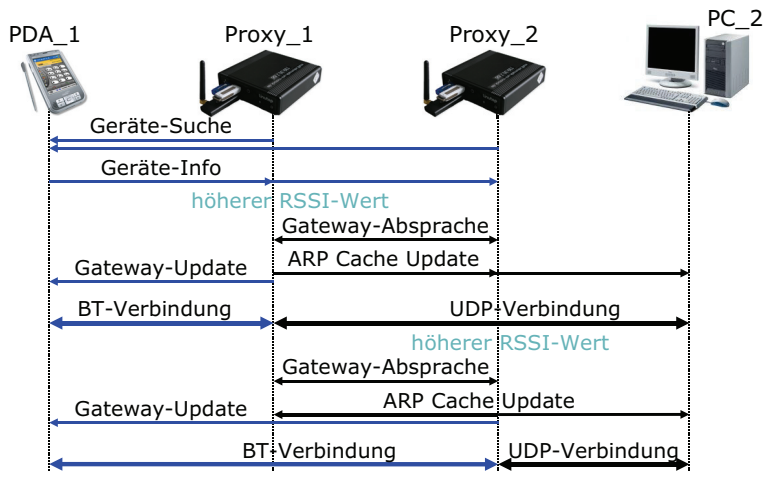

Abbildung 3.27 Sequenzdiagramm für heterogene Adressierung und Mobilität

Wie zuvor in den Abschnitten 3.5.1 und 3.5.2 beschrieben, suchen Proxys regelmäßig nach Bluetooth-Geräten in ihrer Umgebung. Sobald sie den PDA in ihrer Reichweite gefunden haben, können sie Informationen, wie z. B. den aktuellen RSSI-Wert zu ihm bestimmen, der ein Maß für die Qualität der Funkverbindung zwischen beiden angibt. Da sich die Proxys in einem Verbund einer Community befinden und proaktiv Statusinformationen untereinander austauschen, informieren sie auch ihre Nachbarknoten über das mobile Gerät in ihrer Bluetooth-Zelle. Damit befinden sich Proxy_1 und Proxy_2 auf dem gleichen Informationsstand. In die Metrik für eine Handover-Entscheidung können hierbei der RSSI-Wert und die Latenz zum mobilen Gerät einfließen. Weiterhin können die *Signal to Noise Ratio* (SNR), die Paketverlustrate oder die Auslastung der Bluetooth-Schnittstelle der Proxys als Parameter für eine Metrik betrachtet werden. Hierbei sei auf die weiterführende Literatur [108][109] verwiesen, die sich detaillierter mit dem Thema der Handover-Entscheidungen beschäftigt. Da eine derartige Metrik aber nicht zum Kernbestandteil dieser Arbeit gehört, wird an dieser Stelle vereinfacht der RSSI-Wert als Parameter für die Handover-Entscheidung genutzt. Dabei wird angenommen, das am Anfang der RSSI-Wert zwischen Proxy_1 und dem PDA relativ hoch ist und zwischen Proxy_2 und dem PDA niedrig. Eine Absprache zwischen Proxy_1 und Proxy_2 ergibt dann, dass Proxy_1 am besten als Gateway für den PDA geeignet ist. Daraufhin informiert Proxy_1 das IP-Netz mit einem ARP Cache Update, dass der PDA über ihn als Gateway erreichbar ist. IP-basierte Verbindungen an die virtuelle IP-Adresse des PDAs werden somit an den Proxy_1 geleitet, der sie in Bluetooth-Verbindungen an den PDA umsetzt. Mit der Bewegung des PDAs an den Rand der Bluetooth-Zelle von Proxy_1 in Richtung Proxy_2, verringert sich der RSSI-Wert zu Proxy_1 und der RSSI-Wert zu Proxy_2 wird erhöht. Wieder findet eine Absprache zwischen den Proxys statt. Diesmal ist die Funkverbindung von Proxy_2 zum PDA deutlich besser als die von Proxy_1 und Proxy_2 muss die Gateway-Funktionalität für den PDA übernehmen. Dazu führt nun Proxy_2 einen ARP Cache Update im IP-Netz durch, wodurch IP-basierte Verbindungen an die virtuelle IP-Adresse des PDAs nun über Proxy_2 umgeleitet und in Bluetooth-Verbindungen an den PDA umgesetzt werden. Ein derartiges Handover sollte jedoch nur bei einer zu erwartenden deutlich besseren Verbindung (hier bei einem deutlich besseren RSSI-Wert) durchgeführt werden, um durch Hysterese Oszillationen beim Wechsel der Proxys zu vermeiden.

Beim Handover selbst gibt es verschiedene Ansätze, in welchem Maße die beteiligten Geräte Einfluss darauf haben und welches Gerät das Handover schließlich initiiert. Beim ersten Ansatz wählt der End-Knoten, also das mobile Gerät, sein Gateway selber aus einer Liste der in der nähe befindlichen Gateways aus, initiiert das Handover und informiert die beteiligten Gateways (GPAPs). Dabei kann die im Abschnitt 3.5.2 beschriebene Auswahl eines BT-IP-Gateways durch das mobile Gerät ange-

wendet werden, bei dem Gateways ihre Funktionalität als SDP-Dienste im lokalen Bluetooth-Netz bereitstellen. Im zweiten Ansatz nutzt der Gateway-Verbund seine Informationen über lokale Bluetooth-Geräte für die Entscheidung eines Handovers. Hierbei findet eine Absprache der Gateways untereinander statt, die z. B. die aktuellen RSSI-Werte und Latenzen zu den Geräten sowie die Auslastung der Bluetooth-Schnittstellen der Proxys berücksichtigen können, um festzustellen ob bzw. zu welchem Gateway eine Verbindungsübergabe sinnvoll ist und initiiert werden sollte. Anschließend wird das mobile Gerät über sein aktuelles Gateway informiert. Der letzte Ansatz ergibt sich aus den beiden vorherigen Ansätzen, indem der Gateway-Verbund und das mobile Gerät in die Handover-Entscheidung eingebunden werden und das Handover gemeinsam initiieren.

Aus Sicht der MAC-basierten Kommunikation des Bluetooth-Gerätes wird beim Handover von Proxy_1 zu Proxy_2 auf der Ebene der heterogenen Adressierung also nur die MAC-Adresse des Default-Gateways geändert und die für das IP-Netz bestimmten Daten an das neue Gateway gesendet. Aus Sicht der IP-basierten Geräte ändert sich mit dem durch Proxy_2 initiierten ARP Cache Update automatisch die MAC-Adresse des nächsten Hops auf dem Weg zum Bluetooth-basierten Gerät. Die virtuelle IP-Adresse des entfernten Bluetooth-Gerätes bleibt hingegen gleich, was die Kommunikation auf höheren ISO-OSI-Schichten (z. B. bei der Dienstnutzung) deutlich erleichtert. Auch Änderungen beim DHCP- oder DNS-Server entfallen, da die virtuelle IP-Adresse des PDAs und sein Name gleich geblieben sind.

3.6.2 Mobilität auf der Ebene des Service Proxyings

Um die im Mobilitätsszenario in Abbildung 3.26 beschriebene Dienst-Mobilität zu erreichen, muss das Service Proxying wieder aus beiden Blickwinkeln; der IP-basierten Web Services und der MAC-basierten Bluetooth-SDP Dienste, betrachtet werden.

Aus dem Blickwinkel eines IP-basierten End-Knotens stellt Proxy_1 zuerst einen Web Service WS_A unter der virtuellen IP-Adresse IP_A stellvertretend für den Dienst SDP_A des PDAs bereit und registriert ihn beim WS-Broker. Dieser Web Service kann dann durch den PC genutzt werden. Beim Eintreten in die Funkreichweite von Proxy_2 stellt dieser ebenfalls einen stellvertretenden Web Service WS_A mit der gleichen Dienstbeschreibung und dem gleichen Interface unter der virtuellen IP-Adresse IP_A bereit. Aus Sicht des IP-basierten Gerätes wird WS_A also zweimal angeboten. Solange Proxy_1 die bessere Funkverbindung zum PDA besitzt, wird die Dienstnutzung von WS_A auf den Proxy_1 geleitet, da er zuvor, wie im vorherigen Abschnitt beschrieben, durch das ARP Cache Update im IP-Netzwerk bekannt gemacht hat, dass er das Gateway (der nächste Hop) zum PDA darstellt. Mit der

weiteren Bewegung des PDAs in die Bluetooth-Zelle von Proxy_2 wird die Funkverbindung zwischen beiden besser und überwiegt gegenüber der Verbindung zwischen dem PDA und Proxy_1. In diesem Moment tritt das im vorherigen Abschnitt beschriebene Handover in Kraft, das durch einen ARP Cache Update von Proxy_2 dazu führt, dass die Dienstnutzung beider WS_A-Dienste über den Proxy_2 transparent umgeleitet wird. Nimmt die Funkverbindung zwischen dem PDA und Proxy_1 weiter ab, kann Proxy_1 den PDA nicht mehr finden. Daraufhin stellt er den für den PDA stellvertretenden Dienst WS_A im IP-Netz ein und trägt ihn beim WS-Broker aus. Damit ist der Dienst WS_A nur noch einmal im Dienstverzeichnis vorhanden und kann weiterhin über Proxy_2 durch IP-basierte Geräte genutzt werden. Die Mobilität des Bluetooth-Gerätes bzw. seines Dienstes bleibt bei dieser Vorgehensweise für die IP-basierten Geräte transparent.

Aus dem Blickwinkel von MAC-basierten Bluetooth-Geräten stellen die Proxys durch Service Proxying die im IP-Netz verfügbaren Web Services als SDP-Dienste in ihren lokalen Bluetooth-Zellen bereit. Ein PDA, der sich in der Funkzelle von Proxy_1 befindet, kann somit den stellvertretenden Bluetooth-Dienst SDP_B mit der zugehörigen ID (ID_B) unter Verwendung von Proxy_1 als Default-Gateway nutzen. Beim Eintritt des PDAs in die Funkzelle von Proxy_2 steht dem PDA der für den PC stellvertretende Dienst SDP_B zweimal zur Verfügung. An dieser Stelle entscheidet die Einstellung des Default-Gateways darüber, das der von Proxy_1 angebotene Dienst genutzt wird. Mit dem Verlassen der Funkzelle von Proxy_1 wird durch das zuvor beschriebene Handover auf Adressierungsebene auch das Default-Gateway auf dem PDA aktualisiert und die Dienstnutzung über Proxy_2 umgeleitet. Mit dieser Vorgehensweise wird nun auch das transparente Service Proxying von IP-basierten Web Services auf MAC-basierte Bluetooth-SDP-Dienste für mobile Dienstnutzer innerhalb einer Community ermöglicht.

Nachdem in den vorherigen Abschnitten auf die Kombination homogener horizontaler Zellen zu einem heterogenen Ensemble mit Hilfe einer zentralisierten Organisationsform für Gateways eingegangen und mögliche Kommunikationsformen innerhalb einer Community anhand des Service Proxying und der heterogenen Adressierung näher erläutert wurden, widmet sich dieser Abschnitt der Referenzarchitektur eines General Purpose Access Points, der eine allgegenwärtige Kommunikation ermöglicht und die Mobilität der Geräte bzw. deren Dienste innerhalb einer Community unterstützt.

3.7 Referenzarchitektur des General Purpose Access Points

Die Referenzarchitektur des General Purpose Access Points soll die Basis für ein interoperables und effizientes Kommunikationssystem bilden, das die Heterogenität aktu-

eller pervasiver Umgebungen (z. B. Smart Homes, Smart Ensembles) sowohl auf der Service- als auch auf der Netzwerk-Ebene systematisch überwindet und eine transparente Einbindung neuer Geräte ermöglicht. Die Referenzarchitektur des GPAPs basiert auf der in Abschnitt 3.4.2 beschriebenen zentralisierten Gateway-Funktionalität. Dabei besteht der GPAP aus einem eingebetteten System und verfügt über verschiedene Netzwerkschnittstellen, durch die er direkt mit Geräten in seiner Umgebung kommunizieren kann, die ebenfalls über eine der Schnittstellen verfügen. Er ist somit Teilnehmer mehrerer horizontaler Netzwerkstrukturen und verbindet diese zu einem Ensemble. Die Vermittlung zwischen den Zellen wird dabei auf den in Abbildung 3.28 dargestellten Schichten durchgeführt.

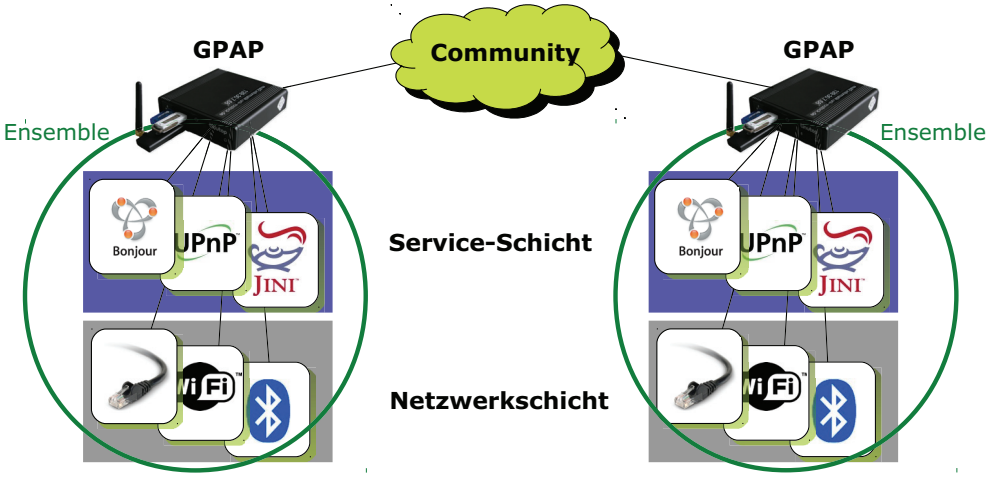

Abbildung 3.28 Kombination aus Netzwerk- und Service-Schicht des GPAPs

Die Service-Schicht ermöglicht die im Abschnitt 3.5.1 beschriebene Service-basierte Kommunikation in heterogenen Netzen in Form des Service Proxying. Die Netzwerkschicht befasst sich dagegen mit der in Abschnitt 3.5.2 beschriebenen adressbasierten Kommunikation für heterogene Ensembles. Durch die Zusammenarbeit beider Schichten und die Kopplung der GPAPs über ein IP-basiertes Backbone kann die im letzten Abschnitt beschriebene Mobilität der End-Knoten und deren Dienste innerhalb einer Community erreicht werden.

3.7.1 Die Service-Schicht des GPAPs

Die Service-Schicht fokussiert die Gerätekooperation auf der Anwendungsebene, auf der die in pervasiven Umgebungen notwendige Flexibilität bei der Auswahl und Nutzung der Geräte mit Hilfe einer Service-orientierten Architektur gewährleistet wird.

Durch die Service-Schicht werden die in den verschiedenen Netzwerktechnologien genutzten und größtenteils miteinander inkompatiblen Lösungen zur Realisierung einer SOA zusammengebracht. Die im Rahmen des Graduiertenkollegs MuSAMA in der Arbeit von Herrn Raphael Zender konzipierte und entwickelte Service-Schicht basiert auf dem Ansatz einer *Service Technology Independent Architecture* (STIA) [102]. Sie nutzt eine *Service Technology Independent Language* (STIL), um konkrete Dienstbeschreibungen in einer abstrakten Form zwischenzuspeichern. STIA besteht aus drei Teilen: dem *STIL Service Manager* (SSM), mehreren *Service Technology Translators* (STTs) und unterschiedlichen *Service Technology Plugins* (STPs). Jedes dieser STPs wird genutzt, um Services einer speziellen Service-Technologie zu finden bzw. bereitzustellen. Die STTs dienen dazu, die durch die STPs gefundenen Dienste in eine abstrakte STIL-Beschreibung zu übersetzen. Dabei können von der Netzwerkschicht bereitgestellte Kontextinformationen genutzt werden, um die Service-Beschreibungen mit QoS-Parametern (wie z. B. Latenzen zwischen dem Endgerät und dem GPAP oder die aktuelle Auslastung der Netzwerkanbindung) anzureichern. Anschließend verwendet der SSM die Service-Beschreibungen um daraus unter Nutzung spezieller STTs Dienste einer anderen Service-Technologie zu generieren und sie mit Hilfe eines speziellen STPs zu veröffentlichen. Diese Transformation der Service-Beschreibungen funktioniert in beiden Richtungen auf die gleiche Weise. Der Einsatz von STIL als Zwischenbeschreibungssprache führt dabei zur Reduktion der benötigten STTs, da hierdurch für die Integration einer neuen Service-Technologie nur ein STP und ein spezieller STT zur Übersetzung in die Zwischenbeschreibungssprache nötig ist, anstatt STTs zu allen anderen Service-Technologien entwickeln zu müssen. Durch einen *Peer-to-Peer* (P2P)-basierten Austausch von STIL-Beschreibungen zwischen den Ensembles einer Community erlaubt es die STIA-Architektur weiterhin, Dienste eines Ensembles nicht nur in andere Dienst-Technologien zu übersetzen und innerhalb desselben Ensembles anzubieten, sondern sie auch in entfernten Ensembles einer Community bereitstellen zu können.

Für eine tiefgreifendere Beschreibung der STIA-Architektur und dessen Plugin-basierten Aufbaus sei an dieser Stelle auf die Arbeiten von Herrn Raphael Zender verwiesen [102][110].

3.7.2 Die Netzwerkschicht des GPAPs

Die Herausforderung der Netzwerkschicht besteht, wie bereits in Abschnitt 3.5.2 beschrieben, in einer für die unterschiedlichen Endgeräte transparenten Adressierung und Kommunikation innerhalb eines Ensembles, die durch einen GPAP (als Gateway bzw. Proxy) ermöglicht wird. Im Rahmen des Graduiertenkollegs MuSAMA wurde mit dieser Arbeit eine Architektur namens *Heterogeneous Context-based*

Routing (HCBR) [98] für einen derartigen GPAP entwickelt, die eine Integration von drahtgebundenen und drahtlosen Technologien jeglicher Art ermöglicht und die Gateway-Funktionalität zwischen unterschiedlichen Netzwerktechnologien übernimmt. Die HCBR-Architektur ist, wie in Abbildung 3.29 dargestellt, in drei Ebenen aufgeteilt und arbeitet ähnlich wie ein Routing-Daemon auf den GPAPs der Community.

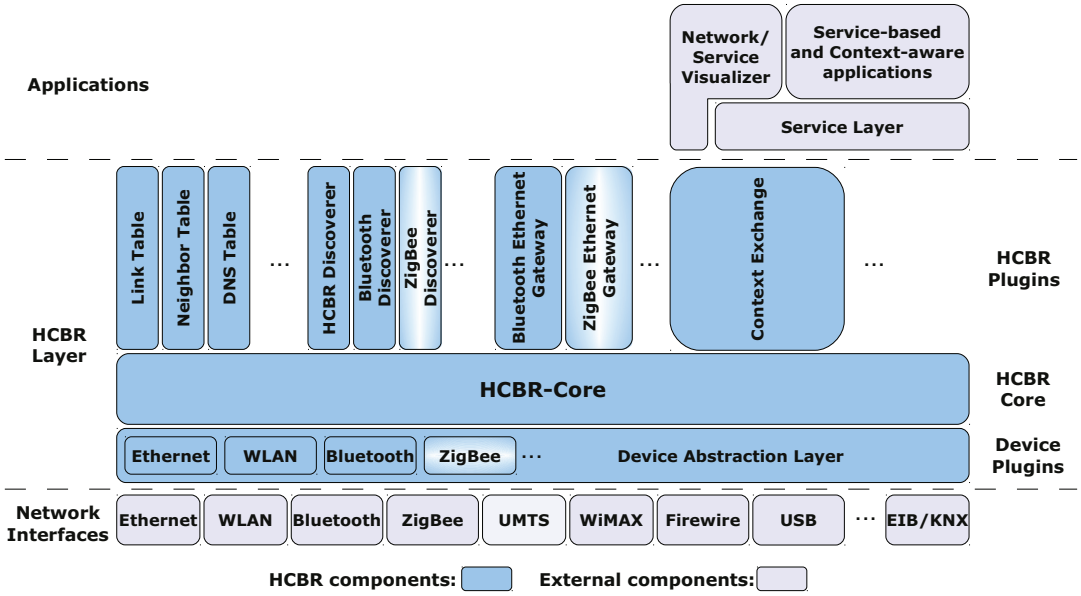

Abbildung 3.29 Aufbau der HCBR-Architektur

Der *HCBR-Core* stellt den zentralen Bereich der Netzwerkschicht eines GPAPs dar. Mit Hilfe eines Plugin-Konzepts ist er modular mit *Device-Plugins* und *HCBR-Plugins* erweiterbar und kann so beliebig um die Unterstützung neuer Netzwerktechnologien und Funktionen erweitert werden. Dabei ist er nicht nur für das Laden, Entfernen, Starten und Stoppen der Device- und HCBR-Plugins zuständig, sondern auch für deren Zusammenarbeit. Dazu können HCBR-Plugins angeben, welche Device-Plugins sie verwenden und welche Funktionen sie anderen HCBR-Plugins zur Verfügung stellen möchten. Diese Funktionen werden dann über spezielle Interfaces beim HCBR-Core registriert und anderen HCBR-Plugins auf Nachfrage zugänglich gemacht. Weiterhin kümmert sich der HCBR-Core um das Logging von Zustandsinformationen der Netzwerkschicht und vereinfacht so das Debugging des Netzwerkverhaltens zur Laufzeit.

Device-Plugins stellen die Schnittstelle des HCBR-Cores zu den im Betriebssystem vorhandenen Netzwerkschnittstellen dar. Dazu ist für jede neue Netzwerktechnologie ein spezielles Device-Plugin notwendig, das die individuellen Charakteristiken beim

Zugriff auf die drahtgebundene oder drahtlose Netzwerkschnittstelle berücksichtigt und dem HCBR-Core eine standardisierte Schnittstelle bereitstellt. Dabei kann z. B. im Ethernet-Device-Plugin die *libpcap*-Bibliothek [111] für einen vereinfachten lesenden Zugriff oder die *libnet*-Bibliothek [112] für einen vereinfachten schreibenden Zugriff auf die Netzwerkschnittstelle eingesetzt werden. Ein Bluetooth-Device-Plugin nutzt dagegen eher die *BlueZ*-Bibliothek [113] für diese Aufgaben. Device-Plugins bieten dem HCBR-Core also einerseits Informationen über die Netzwerkschnittstelle und das Plugin selbst, wie z. B. dessen Typ, Version und Autor. Andererseits stellen sie standardisierte Methoden zum Initialisieren sowie wieder Freigeben belegter Ressourcen und zum Lesen sowie Schreiben von Datenpaketen über eine Netzwerkschnittstelle bereit. Zur Realisierung des im vorherigen Abschnitt beschriebenen zentralisierten GPAPs werden Device-Plugins für Ethernet, WLAN, Bluetooth und ZigBee benötigt. Weiterhin können aber auch Device-Plugins in Form von Datenpaket-Generatoren für Untersuchungen und Tests im Rahmen der HCBR-Architektur implementiert und genutzt werden.

HCBR-Plugins werden dazu verwendet, den HCBR-Core um neue Funktionalitäten zu erweitern. Dabei greifen sie nicht direkt auf die im Betriebssystem vorhandenen Netzwerkschnittstellen zu, sondern nutzen die durch den HCBR-Core verwalteten Device-Plugins. Die Aufgaben der HCBR-Plugins lassen sich grob in 3 Bereiche aufteilen, die in den folgenden Abschnitten detaillierter dargestellt werden: a) die Topologie-Bestimmung des GPAP-Verbundes und Kopplung der GPAPs zu einem lokalen Backbone innerhalb einer Community, b) die Realisierung der Gateway-Funktionalität für die horizontalen Zellen durch Realisierung des Ansatzes der heterogenen Adressierung und c) die Schnittstelle zur Service-Schicht.

a) Topologie-Bestimmung des GPAP-Verbundes

Für die Kopplung der GPAPs zu einem lokalen Backbone eines heterogenen Netzwerks wird, wie schon in Abschnitt 3.4.3 beschrieben und in Abbildung 3.13 dargestellt, auch hier davon ausgegangen, dass sich in einer aus mehreren Umgebungen bestehenden Community in jeder Umgebung ein GPAP befindet und diese drahtgebunden (z. B. per Ethernet) oder drahtlos (z. B. per WLAN) miteinander verbunden sind. In Anlehnung an den OLSR-Daemon [25] für mobile Ad-hoc-Netzwerke nutzt das HCBR-Discoverer-Plugin die Netzwerkschnittstelle zum Backbone, um regelmäßig Hello-Nachrichten in Form von Broadcasts an die am Backbone beteiligten GPAPs zu senden und sich dadurch im Backbone bekannt zu machen. Diese Hello-Nachrichten enthalten unter anderem den Namen des Gerätes, die Anzahl und die Typen seiner Netzwerkschnittstellen und einen *Globally Unique Identifier* (GUID). Gleichzeitig nutzen die GPAPs im Backbone ihr HCBR-Discoverer-Plugin um Hello-Nachrichten zu empfangen und so ihre benachbarten GPAPs zu ermitteln. Von diesen Informationen wird der GUID mit seinen dazugehörigen Netzwerkschnittstellen

in die vom Neighbor-Table-Plugin bereitgestellte Nachbar-Tabelle eingetragen. Die Verbindung zwischen dem lokalen GPAP und dem GPAP, der die Hello-Nachricht versendet hat, wird in der vom Link-Table-Plugin realisierten Link-Tabelle vermerkt. In die DNS-Tabelle des DNS-Table-Plugins wird anschließend der GUID und der dazugehörige Name hinterlegt. Auf die Funktionalität der Neighbor-, DNS- und Link-Table-Plugins kann das HCBR-Discoverer-Plugin dabei unter Nutzung ihrer jeweils zuvor beim HCBR-Core registrierten Interfaces zurückgreifen. Durch die Zusammenarbeit der vier beschriebenen HCBR-Plugins kann somit eine grundlegende Topologie-Bestimmung des GPAP-Verbundes erreicht werden.

Damit aber nicht nur GPAPs sondern auch die übrigen Geräte bei der Topologie-Bestimmung berücksichtigt werden, sind weitere HCBR-Plugins notwendig. Dazu gehört z. B. das Bluetooth-Discoverer-Plugin, das das Bluetooth-Device-Plugin nutzt, um Bluetooth-Geräte in der lokalen Umgebung des GPAPs zu finden. Dabei werden durch regelmäßige Inquirys die in der lokalen Bluetooth-Zelle vorhandenen Geräte gefunden, die Verbindung zwischen GPAP und Bluetooth-Gerät in die Link-Tabelle (die IP-Adressen des Links bleiben hierbei leer) und der Name des Gerätes in die DNS-Tabelle aufgenommen. Hierbei wird wieder auf die zuvor beim HCBR-Core registrierten Interfaces der Neighbor-, DNS- und Link-Table-Plugins zurückgegriffen. Als Alternative zum regelmäßigen Inquiry bietet sich im Bluetooth-Discoverer-Plugin auch die Nutzung des *DBus* [114] zur Suche nach lokalen Bluetooth-Geräten an, da der Bluetooth-Daemon der unter Linux am weitesten eingesetzten Bluetooth-Bibliothek BlueZ über den DBus ansprechbar ist [115]. Dabei ist es notwendig, dass eine Bluetooth-Schnittstelle des GPAPs durch das Bluetooth-Discoverer-Plugin am DBus angemeldet und für das Empfangen von sogenannten Signalen registriert ist. Mit einem Funktionsaufruf über den DBus an den Bluetooth-Daemon kann dann ein regelmäßiges Inquiry eingeleitet werden. In der anderen Richtung bekommt das Bluetooth-Discoverer-Plugin Änderungen in der Bluetooth-Zelle vom Bluetooth-Daemon über die zuvor registrierten Signale mitgeteilt. Durch die Auswertung des Signals „RemoteDeviceFound“ kann die MAC-Adresse und der RSSI-Wert zu einem in der Nähe befindlichen Bluetooth-Gerät ermittelt und die Informationen in die entsprechenden Tabellen (HCBR-Plugins) eingetragen werden. Der RSSI-Wert kann z. B. als Pfadgewicht mit in der Link-Tabelle gespeichert werden. Die Auswertung des Signals „RemoteNameUpdate“ liefert weiterhin den Namen des Gerätes und wird ebenfalls in der DNS-Tabelle hinterlegt.

In Anlehnung an das Bluetooth-Discoverer-Plugin kann auch ein ZigBee-Discoverer-Plugin realisiert werden, das die lokalen ZigBee-Geräte mit in die Topologie-Bestimmung des GPAP-Verbundes aufnimmt. Auf diese Weise können die mobilen Bluetooth- und ZigBee-Geräte in die Link-, Neighbor- und DNS-Tabellen eines GPAPs eingetragen und in die Topologie des Backbone-Verbundes integriert werden.

Dadurch, dass das HCBR-Discoverer-Plugin nicht nur Hello-Nachrichten an die Nachbarn im GPAP-Verbund schickt, sondern durch Nutzung von Callback-Funktionen der speziellen Interfaces der Link-, Neighbor- und DNS-Table-Plugins auch Änderungen in deren Tabellen mitbekommt und daraufhin die Nachbarn über diese Änderungen informiert, werden die Topologie-Informationen auf den GPAPs im Verbund verteilt. Auf diese Weise kennt jeder GPAP im Verbund die heterogene Topologie der Community, die aus den GPAPs und deren lokal erreichbaren mobilen Geräten besteht. Dies stellt eine fundamentale Voraussetzung für die weitere Vermittlerfunktionalität der GPAPs, die heterogene Adressierung, die Unterstützung der Mobilität durch ein heterogenes Handover sowie die in aufbauenden Arbeiten zu entwickelnden Metriken zum Routing in heterogenen Netzwerken dar.

b) Realisierung der Gateway-Funktionalität für horizontale Zellen
Nach der Realisierung einer gemeinsamen Topologie eines heterogenen Netzes kommt der Realisierung der in Abschnitt 3.5.2 am Beispiel der Kombination von Ethernet- und Bluetooth-Zellen beschriebenen Gateway-Funktionalität eine große Bedeutung in der HCBR-Architektur zu. Der Ansatz der heterogenen Adressierung; mit der Bereitstellung einer virtuellen IP-Adresse für mobile Bluetooth-Geräte und der Bekanntmachung im IP-Netzwerk über welches Gateway (bzw. welchen GPAP) das Gerät momentan erreichbar ist, wird in der HCBR-Architektur durch ein Plugin namens Bluetooth-Ethernet-Gateway bereitgestellt und im folgenden Abschnitt detaillierter vorgestellt.

Bei der Initialisierung des Plugins muss zunächst überprüft werden, welche Ethernet-Schnittstelle als Schnittstelle zum Backbone bereitsteht und somit einen dort vorhandenen DHCP-Server für die spätere Vergabe der virtuellen IP-Adressen für Bluetooth-Geräte nutzen kann. Die Suche der in der Umgebung des GPAPs befindlichen Bluetooth-Geräte wird wie im vorherigen Abschnitt erläutert, durch das Bluetooth-Discoverer-Plugin übernommen, das aktuelle Verbindungen in der Link-Tabelle speichert. Durch Nutzung einer Callback-Funktion des Link-Table-Plugins kann das Bluetooth-Ethernet-Gateway-Plugin auf eine recht elegante Weise über Änderungen in der Link-Tabelle und damit auch auf neu gefundene oder nicht mehr zu findende Bluetooth-Geräte informiert werden und darauf entsprechend reagieren. Lokal vorhandene Bluetooth-Verbindungen sind dabei in der Link-Tabelle mit dem Schnittstellen-Typ *BLUETOOTH* eingetragen und als lokal markiert. Für jedes neue lokale Bluetooth-Gerät wird nun eine virtuelle IP-Adresse beim DHCP-Server im Backbone angefragt. Genau genommen wird an dieser Stelle der lokale *DHCP Client Daemon* (DHCPCD) auf dem GPAP verwendet und zu der Ethernet-Schnittstelle eine Dummy-Schnittstelle mit einem freien Alias und einer Dummy-IP auf dem GPAP angelegt (z. B. per ifconfig eth0:x 127.0.0.x), die einem bestimmten Bluetooth-Gerät zugeordnet ist. Mit Hilfe dieser Dummy-Schnittstelle und der

MAC-Adresse des Bluetooth-Gerätes führt der DHCPCD anschließend ein IP-Lease beim DHCP-Server durch und bekommt ein IP-Offer zurück, mit dem der Dummy-Schnittstelle eine IP-Adresse zugewiesen wird. Diese IP-Adresse entspricht nun der virtuellen IP-Adresse eines bestimmten Bluetooth-Gerätes und kann vom Bluetooth-Ethernet-Gateway-Plugin aus einem vom DHCPCD angelegten Info-File ausgelesen und lokal im genutzten Link-Eintrag der Link-Tabelle gespeichert werden. Zusätzlich wird ein ARP Cache Update mit der virtuellen IP-Adresse und der MAC-Adresse der Ethernet-Schnittstelle des GPAPs im Backbone durchgeführt, um die IP-basierten Geräte darüber zu informieren, das die virtuelle IP-Adresse über das entsprechende Gateway erreichbar ist. Bekommt das Bluetooth-Ethernet-Gateway-Plugin erneut die Benachrichtigung vom Link-Table-Plugin, dass das Bluetooth-Gerät in der lokalen Reichweite liegt, so wird überprüft, ob das IP-Lease noch gültig ist und gegebenenfalls erneuert werden muss. Bei Benachrichtigung, dass ein Link zum Bluetooth-Gerät veraltet ist, wird die entsprechende Dummy-Schnittstelle bei der Ethernet-Schnittstelle gelöscht und das zugehörige Info-File wieder entfernt.

Um die Kommunikation zwischen IP- und nicht-IP-basierten Geräten zu ermöglichen, setzt das Bluetooth-Ethernet-Gateway-Plugin die Bibliothek libnetfilter_queue [116] ein, die eine konditionale Verarbeitung und Manipulation von eingehenden sowie ausgehenden Datenpaketen ermöglicht, bevor sie vom Betriebssystem-Kern verarbeitet werden. Diese Datenpakete werden in einer Queue hinterlegt und von einem Userspace-Programm abgearbeitet. Hierbei wird entweder nur der Paket-Header (mit Ausnahme des Ethernet-Headers) oder der gesamte Payload zur Analyse aus dem Kernelspace in den Userspace kopiert, dort manipuliert und nachher wieder in den Kernelspace zurück gesendet. Diese Funktionalität erlaubt es z. B. ICMP Echo-Request Pakete, die von IP-basierten Geräten im Backbone an die virtuelle IP-Adresse eines Bluetooth-Gerätes geschickt werden, abzufangen und an das Bluetooth-Gerät in Form eines Layer-2-Pings weiterzuleiten. Gleichzeitig werden zurückgehende ICMP-Echo-Reply Pakete solange verzögert, bis die Antwort vom Layer-2-Ping wieder eingetroffen ist. Auf gleiche Weise werden aber auch am GPAP eingehende UDP- und TCP-Daten an die virtuelle IP-Adresse eines lokalen Bluetooth-Gerätes verarbeitet und im Bluetooth-Netz über RFCOMM-Verbindungen an das zugehörige Bluetooth-Gerät weitergeleitet.

Mit der Mobilität der Bluetooth-Geräte innerhalb einer Community kommt auf dieses Plugin aber auch das in Abschnitt 3.6 beschriebene Verfahren zur Unterstützung der Mobilität auf der Ebene der heterogenen Adressierung zu. Dabei werden die in der Link-Tabelle gespeicherten RSSI-Werte als ein einfaches Maß zur Qualität der Funkverbindung zwischen den GPAPs und den mobilen Geräten verwendet, um nach einer durch mehrere GPAPs ausgehandelten Handover-Entscheidung die Gateway-Funktionalität an einen für das mobile Geräte besser geeigneten GPAP ab-

zugeben. Hierbei können auch statistische Verfahren oder Kontextinformationen aus der Service-Schicht genutzt werden, um die Mobilität der Geräte teilweise vorauszusagen und somit schneller auf deren Mobilität reagieren zu können. Wurde ein besser geeigneter GPAP gefunden, übernimmt er die Gateway-Funktionalität für das jeweilige Bluetooth-Gerät und führt einen ARP Cache Update im IP-Netz durch, wodurch IP-basierte Verbindungen an die virtuelle IP-Adresse des mobilen Gerätes über den neuen GPAP umgeleitet werden. Mit dieser Vorgehensweise wird ein transparenter Gateway-Wechsel für die aktuellen Verbindungen (Sockets) zwischen dem IP-basierten Backbone und den nicht-IP-basierten Bluetooth-Zellen ermöglicht.

In Anlehnung an das Bluetooth-Ethernet-Gateway-Plugin wird für die transparente Einbindung von ZigBee-Geräten in das IP-Netz ein ZigBee-Ethernet-Gateway-Plugin benötigt, damit sich nicht-IP-basierte ZigBee-Geräte und IP-basierte Geräte gegenseitig adressieren und miteinander kommunizieren können. Um mit der steigenden Anzahl an neuen Netzwerktechnologien die Anzahl der benötigten Gateway-Plugins ebenfalls nur linear ansteigen zu lassen, wird Ethernet für die Gateway-Plugins als Basis-Technologie verwendet und es werden nur Gateway-Plugins zwischen der Basis-Technologie und der jeweils neuen Technologie implementiert. Auf diese Weise werden Gateways für die Kombination von Ethernet-Bluetooth und Ethernet-ZigBee benötigt, jedoch nicht für Bluetooth-ZigBee, da diese Funktionalität durch die Kombination der beiden vorherigen Gateway-Plugins realisiert werden kann.

c) Kombination von Netzwerk- und Service-Schicht

Die dritte Art der HCBR-Plugins dient als Schnittstelle für die Zusammenarbeit der Netzwerk- und der Service-Schicht innerhalb eines GPAPs. Ein Vertreter dieser Art von Plugins ist das Context-Exchange-Plugin, das für den Datenaustausch zwischen der Netzwerkschicht und der Service-Schicht verantwortlich ist. Beim Datenaustausch wird eine *Inter-Process Communication* (IPC) in Form eines verbindungsorientierten TCP-Sockets genutzt, um Informationen innerhalb des GPAPs zwischen zwei Schichten auszutauschen, deren Speicherbereiche strikt voneinander getrennt sind. Durch die IPC-Verbindung ist es unter Nutzung von Funktionsaufrufen und Signalen der jeweils anderen Schicht möglich, auf Veränderungen des Kontextes der Umgebung oder der darin enthaltenen Services zu reagieren. Somit wird der Service-Schicht ein lesender Zugriff auf die HCBR-Konfiguration und die aktuell verwendeten Device- und HCBR-Plugins der Netzwerkschicht ermöglicht, um Kontextinformationen über vorhandene Routen in der Community mit ihren Parametern Latenz, verfügbare/genutzte Bandbreite oder der Qualität der Funkverbindung sowie deren Änderungen aus der Link-, DNS- und Neighbor-Tabelle mitverfolgen zu können und in der Service-Schicht zur dynamischen Anreicherung von Dienstbeschreibungen zu nutzen. In der anderen Richtung können aber auch Informationen aus der Service-Schicht genutzt werden, um z. B. Bewegungsinformationen eines Ge-

rätes aus dem Service-basierten Weltmodell einer Community zu erhalten und das Vorwissen, wie sich das Ensemble in der nächsten Zeit räumlich verändern wird, an die Gateway-Plugins weitergeben und dort für die Verbesserung des Routings und des Handovers zu nutzen. Auch ist es denkbar, diese Informationen in einem späteren Routing-Plugin für die Optimierung der Netzwerkpfade zwischen bestimmten Geräten für bestimmte Dienste mit Echtzeitanforderungen, wie z. B. einem heterogenen Video- oder Audio-Chat, nutzen zu können. Während der Entwicklung der bisherigen HCBR-Architektur wurde das Context-Exchange-Plugin aber auch dazu verwendet, die Informationen der Link-, DNS-, und Neighbor-Tabellen an eine Java-basierte grafische Oberfläche namens *Network/Service Visualizer* zu übertragen, die die aktuellen Netzwerkinformationen grafisch aufbereitet in einem Netzwerkgraphen darstellt und so eine bessere Analyse des aktuellen GPAP-Verbundes ermöglicht.

Neben den vorgestellten Arten an HCBR-Plugins sind in weiterführenden Arbeiten aber auch Plugins realisierbar, die das Netzwerkverhalten in der Community möglichst positiv beeinflussen. Dazu zählt unter anderem ein Routing-Plugin, das die Topologie-Informationen des GPAP-Verbundes, die Auslastung der beteiligten Netzwerkschnittstellen und aktuelle Kontextinformationen aus der Service-Schicht für die Routen-Berechnung in einem heterogenen Netzwerk verwendet, um die Netzwerkkapazität in der Community zu erhöhen und möglichst gut zu nutzen. Dabei können messbasierte Verfahren [117] und statusbasierte Verfahren [118] z. B. für ein Multi-Path-Routing oder die Nutzung mehrerer Übertragungskanäle innerhalb eines größeren WLANs [119][120][121] mit unterschiedlichen, möglichst nicht überlappenden Funkkanälen betrachtet und eingesetzt werden.

Ebenfalls bietet sich die in Abschnitt 3.2 beschriebene Betrachtung der logischen Konnektivität des GPAP-Verbundes (in Form von Vermittlerfunktionalitäten durch vorhandene xxx-Ethernet-Gateway-Plugins der beteiligten GPAPs) zur Einschätzung der Konnektivität sowie als Orientierungshilfe und Ansatzpunkt für eine mögliche Erhöhung der Kapazität des heterogenen Netzwerks an.

3.8 Zusammenfassung

In diesem Kapitel der vorliegenden Arbeit wurde das Konzept und die theoretischen Grundlagen einer allgegenwärtigen Kommunikation in heterogenen Netzwerken intelligenter Umgebungen vorgestellt. Dazu wurde zuerst die Kommunikation in heterogenen Netzwerken mit besonderem Blick auf horizontale und vertikale Netzwerkstrukturen sowie deren Kombination zu einem heterogenen Ensemble dargestellt. Mit der anschließenden theoretischen Betrachtung der logischen Konnektivität derartiger Ensembles konnte die große Bedeutung der enthaltenen Gateway-Knoten herausgestellt

werden. Darauf aufbauend wurde auf mögliche zentrale und dezentrale Organisationsformen für Gateways eingegangen, die als Vermittler zwischen den verschiedenen Netzwerktechnologien eingesetzt werden können. Anschließend stellte dieses Kapitel die in heterogenen Netzwerken möglichen Kommunikationsformen anhand des Ansatzes des Service Proxying und der heterogenen Adressierung genauer vor, um schließlich durch Kombination beider Kommunikationsformen der Mobilität aktueller Endgeräte und deren Dienste innerhalb einer Community Rechnung zu tragen. Am Ende dieses Kapitels wurde die Referenzarchitektur für den Aufbau der Software des General Purpose Access Points vorgestellt und detailliert beschrieben. Damit bietet dieses Kapitel die Basis der im nächsten Kapitel folgenden Evaluation der implementierten GPAP-Architektur, die die Kommunikationskonzepte des Service Proxying und der heterogenen Adressierung in verschiedenen Szenarien aufgreift.

Kapitel 4

Realisierung und Evaluation der GPAP-Referenzarchitektur

Bevor dieses Kapitel die untersuchten Evaluationsszenarien der GPAP-Referenzarchitektur vorstellt, soll an dieser Stelle ein kleiner Einblick in die Hardware- und die Software-Umgebung des GPAPs gegeben werden, die die Realisierung der Service- und der Netzwerkschicht des GPAPs maßgeblich beeinflussen.

Als **Hardware-Umgebung** wurden für die GPAPs sogenannte *Linutop2*-PCs [101] eingesetzt. Diese in Abbildung 4.1 dargestellten Mini-Linux-PCs bestehen aus einem eingebetteten System und sind bei einer Größe von gerade einmal 14x14x3,5 cm und einer Leistungsaufnahme von unter 8 Watt, mit einem AMD Geode LX800 Prozessor (x86/32 Bit), 512 MB RAM, 1 GB Flash-Speicher und einer 100 Mbit/s Ethernet-Schnittstelle ausgestattet. Die gewünschte Erweiterbarkeit um neue Netzwerkschnittstellen kann bei dieser Hardwareplattform durch vier USB 2.0-Schnittstellen gewährleistet werden, wodurch WLAN-, Bluetooth- und ZigBee-Schnittstellen in beliebiger Kombination hinzugefügt werden können. Der zusätzlich vorhandene VGA-Ausgang ermöglicht die Entwicklung und das Testen der GPAP-Software direkt auf dem Gerät. Für den Einsatz der GPAPs in den Evaluationsszenarien ist er aber nicht unbedingt notwendig.

Abbildung 4.1 Referenz-Hardware des GPAPs

Als Grundlage der **Software-Umgebung** der GPAPs wird ein Ubuntu-Linux-Betriebssystem in der Version 6.10 eingesetzt, das entweder von einem internen Flash-

Speicher oder einem externen USB-Stick gestartet werden kann. Das Linux-Betriebssystem bietet die Möglichkeit der einfachen Erweiterung um neue Softwarekomponenten. So können z. B. das *Java Development Kit* (JDK) oder die *GNU Compiler Collection* (GCC) in den jeweils aktuellen Versionen installiert werden, um Compiler für die Programmiersprachen C und Java sowie die für den Einsatz von Java benötigte *Java Runtime Environment* (JRE) nutzen zu können.

Bei der Entwicklung der Referenzarchitektur des GPAPs wurden mit C und Java unterschiedliche Programmiersprachen eingesetzt, da sie für die beiden Schichten unterschiedlich gut geeignet sind. Die Service-Schicht setzt auf Java als Programmiersprache, da sie zum einen durch ihre Plattformunabhängigkeit leicht auf andere Betriebssystemplattformen portiert werden kann; zum anderen handelt es sich bei Java um eine bereits breit eingesetzte Programmiersprache im Bereich der Service-orientierten Architekturen, da viele der aktuellen SOA-Technologien eine Java-API anbieten und es sogar einige SOA-Technologien wie z. B. Jini gibt, die nur mit Java nutzbar sind.

Die Netzwerkschicht setzt dagegen auf die Programmiersprache C, da sie einen direkteren Zugriff auf die vom Betriebssystem bereitgestellten Netzwerkschnittstellen und spezielle Betriebssystemfunktionen bietet. Dabei ergeben sich z. B. bei der Nutzung der zuvor beschriebenen Bibliotheken libnet und libpcap Möglichkeiten der Erfassung und der Manipulation von Datenpaketen, was gerade für Device- und HCBR-Plugins, wie z. B. dem Bluetooth-Ethernet-Gateway-Plugin beim Empfang, der Manipulation und der Weiterleitung von Datenpaketen von enormer Bedeutung ist. Durch die Nähe zum Betriebssystem sind mit dem Einsatz der Programmiersprache C aber auch Funktionen des Linux-Kernels selbst nutzbar. So kann im Bluetooth-Ethernet-Gateway-Plugin z. B. das netfilter_queue-Plugin des Linux-Kernels dazu genutzt werden, Datenpakete, die an eine spezielle virtuelle IP-Adresse eines Bluetooth-Gerätes adressiert sind, in eine Warteschlange zu hinterlegen, die dann vom Gateway-Plugin als Userspace-Programm abgearbeitet (akzeptiert, verworfen oder verändert) werden können. Mit weiterführenden HCBR-Plugins, die sich z. B. mit dem Routing in heterogenen Netzwerken oder dem Handover beschäftigen, bietet die Nutzung der Programmiersprache C und des Linux-Betriebssystems aber auch die Möglichkeit, direkt in das Routing-Verhalten des Betriebssystems einzugreifen und dieses aus einem HCBR-Plugin heraus zu steuern. Damit kann in der Netzwerkschicht durch die Nutzung der Programmiersprache C auf viele Funktionen des Betriebssystems zurückgegriffen werden, die aus der Programmiersprache Java heraus nur teilweise oder über umständliche Wege wie z. B. *Java Native Interfaces* (JNI) nutzbar wären.

Als Schnittstelle zwischen der Java-basierten Service-Schicht und der C-basierten Netzwerkschicht wurde mit dem Context-Exchange-Plugin der Netzwerkschicht eine

RPC-Schnittstelle entwickelt, auf die im nächsten Abschnitt noch genauer eingegangen wird.

4.1 Realisierung und Evaluation der HCBR-Architektur

Die im Abschnitt 3.7.2 vorgestellte Netzwerkschicht des GPAPs wurde wie im vorherigen Abschnitt beschrieben, in der Programmiersprache C auf Basis eines Linux-Betriebssystems realisiert. Dabei wurden die Device- und HCBR-Plugins als dynamisch ladbare *Shared Object* (*.so)-Bibliotheken implementiert, die eine modulare und dynamische Erweiterung des HCBR-Cores und somit dem zentralen Bereich der Netzwerkschicht zur Laufzeit ermöglichen. Auf diese Weise kann der HCBR-Core Device- und HCBR-Plugins nicht nur zur Laufzeit laden und starten, sondern auch wieder stoppen, entfernen oder aktualisieren. Für die Zusammenarbeit der HCBR-Plugins teilen diese dem HCBR-Core beim Laden mit, welche Funktionen sie anderen Plugins über spezielle Interfaces bereitstellen können. Diese Interfaces werden beim HCBR-Core registriert und anderen HCBR-Plugins auf Nachfrage zugänglich gemacht. Weiterhin kümmert sich der HCBR-Core um das Logging von Zustandsinformationen der Netzwerkschicht und vereinfacht so das Debugging des Netzwerkverhaltens zur Laufzeit. Die Log-Informationen werden in der Datei */var/log/syslog* gespeichert und können über den Befehl *tail -f /var/log/syslog* mitverfolgt werden.

Als Schnittstelle des HCBR-Cores zu den im Betriebssystem vorhandenen Netzwerkschnittstellen wurden Device-Plugins für Ethernet und Bluetooth realisiert. Dabei wurde im *Ethernet-Device-Plugin* auf die libpcap-Bibliothek [111] für einen vereinfachten lesenden Zugriff und auf die libnet-Bibliothek [112] für einen vereinfachten schreibenden Zugriff auf die Ethernet-Schnittstellen zurückgegriffen. Beim *Bluetooth-Device-Plugin* wurde dagegen die BlueZ-Bibliothek [113] für diese Aufgaben eingesetzt. Aus Sicht der Nutzung der Bibliotheken libpcap und libnet, kann für die Unterstützung der WLAN-Technologie auf das Ethernet-Device-Plugin zurückgegriffen werden, da sich beide Technologien nur auf der Netzzugangsschicht des TCP/IP-Referenzmodells unterscheiden und die gleichen Protokolle der TCP/IP-Familie nutzen. Die Realisierung eines ZigBee-Device-Plugins ist derzeit noch in Arbeit.

Für die Erweiterung des HCBR-Cores um neue Funktionalitäten, wurden mehrere HCBR-Plugins entwickelt. So wurden zuerst die schon in Abschnitt 3.7.2 näher beschriebenen HCBR-Plugins *Link-Table*, *Neighbor-Table* und *DNS-Table* realisiert, die von den weiteren HCBR-Plugins genutzt werden, um Topologieinformationen des heterogenen Netzwerks zu verwalten. So nutzt z. B. das *HCBR-Discoverer-Plugin* die beim HCBR-Core registrierten Interfaces dieser Plugins, um in der Nähe befindliche GPAPs sowie deren Namen und Netzwerkverbindungen zu ihnen in den jeweiligen

Tabellen zu speichern. Die Zusammenarbeit dieser vier HCBR-Plugins erreicht somit eine grundlegende Topologie-Bestimmung eines GPAP-Verbundes. Durch die Realisierung des *Bluetooth-Discoverer-Plugins*, das das Bluetooth-Device-Plugin für den Zugang zur Bluetooth-Netzwerkschnittstelle nutzt, können Bluetooth-Geräte in der lokalen Umgebung eines GPAPs sowohl durch regelmäßige Inquirys als auch durch die Nutzung des DBus im Linux-Betriebssystem gefunden werden. Die Nutzung des DBus stellte sich bei der Evaluation als die bessere Wahl heraus, da hier nicht nur die Namen und MAC-Adressen der in der Umgebung befindlichen Bluetooth-Geräte, sondern auch der RSSI-Wert, der ein Maß für die Funkverbindung zwischen GPAP und Bluetooth-Gerät darstellt, ermittelt und in der Link-Tabelle als einfaches Pfadgewicht gespeichert werden kann. Dieser Wert kann in einem zukünftigen Routing-Plugin als Parameter für eine Routing-Metrik eines heterogenen Netzwerks genutzt werden. Ein ZigBee-Discoverer-Plugin, das die lokalen ZigBee-Geräte mit in die Topologie-Bestimmung des GPAP-Verbundes aufnimmt, ist ebenfalls noch in der Entwicklung, da es zwar die Netzwerkschicht des GPAPs erweitert, sich die Evaluation der grundlegenden Kommunikationskonzepte jedoch auf Bluetooth als einen Vertreter von WPANs beschränkt.

Ein weiteres realisiertes HCBR-Plugin ist das *Context-Exchange-Plugin*. Es dient als Schnittstelle für die Zusammenarbeit der Netzwerk- und der Service-Schicht innerhalb eines GPAPs. Es ist also vor allem für den Datenaustausch zwischen der C-basierten Netzwerkschicht und der Java-basierten Service-Schicht verantwortlich, für den eine Inter-Prozess Kommunikation in Form eines verbindungsorientierten TCP-Sockets genutzt wird. Somit ist es möglich, Kontextinformationen über vorhandene Routen in der Community mit ihren Parametern Latenz, verfügbare/genutzte Bandbreite, Qualität der Funkverbindung sowie deren Änderungen aus der Link-, der DNS- und der Neighbor-Tabelle mitverfolgen und in der Service-Schicht zur dynamischen Anreicherung von Dienstbeschreibungen nutzen zu können. Bei der Realisierung der Netzwerkschicht des GPAPs wurde das Context-Exchange-Plugin aber auch dazu verwendet, um die aktuellen Informationen der Link-, DNS- und Neighbor-Tabellen an den *Network/Service Visualizer* zu übertragen, der diese Informationen in einer einfachen Java-basierten grafischen Oberfläche visualisiert und so eine bessere Analyse des aktuellen Netzwerkzustandes ermöglicht.

Bei der Visualisierung der Netzwerktopologie in Abbildung 4.2 handelt es sich um die vier GPAPs mit den Namen *Kasai*, *Muli*, *Teterow* und *Nil*, die mit jeweils einer Ethernet- und einer Bluetooth-Schnittstelle ausgestattet sind. Der GPAP *Nil* verfügt zusätzlich über eine zweite Ethernet-Schnittstelle, mit der er die 139.30.7.er und 10.10.10.er Ethernet-Zellen verbindet. Weiterhin ist dargestellt, dass sich zusätzliche Bluetooth-fähige Endknoten in der Funkreichweite von Muli sowie Teterow befinden

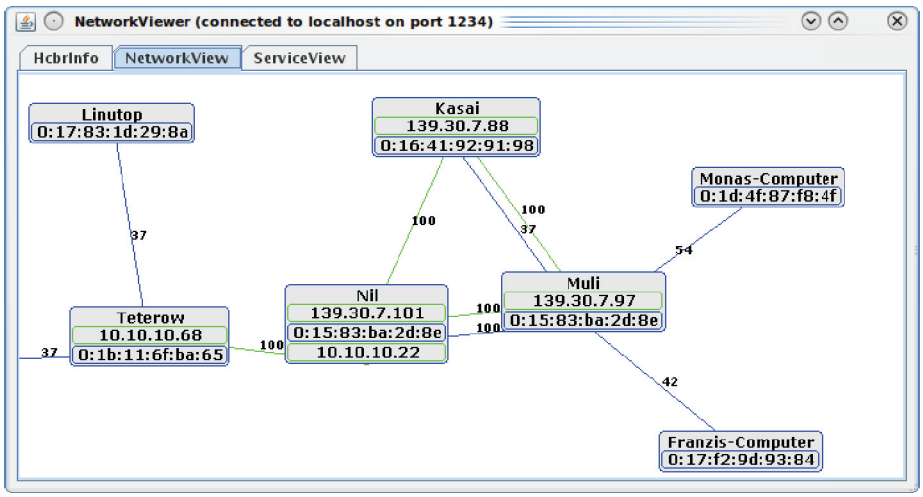

Abbildung 4.2 Darstellung der Netztopologie im Network/Service Visualizer

und der GPAP Muli in der Lage ist, auch über Bluetooth mit Kasai und Nil zu kommunizieren.

4.2 Evaluationsszenario der heterogenen Adressierung

Mit der Realisierung des in Abschnitt 3.7.2 beschriebenen Bluetooth-Ethernet-Gateway-Plugins, welches das Konzept der in Abschnitt 3.5.2 beschriebenen heterogenen Adressierung umsetzt und mobile Bluetooth-Geräte mit Hilfe von virtuellen IP-Adressen für die Endknoten transparent im IP-Netz bekannt und nutzbar macht, konnte eine weitere Aufgabe der Netzwerkschicht des GPAPs evaluiert werden, die gerade für die Kommunikation in heterogenen dynamischen Ensembles von enormer Bedeutung ist. Dabei fügt sich dieses Gateway-Plugin reibungslos als HCBR-Plugin in die HCBR-Architektur der GPAP-Netzwerkschicht ein und kann auf Informationen der Link-, DNS- und Neighbor-Table-Plugins zurückgreifen. So nutzt es z. B. die Callback-Funktionen des Link-Table-Plugins, um über neu gefundene oder nicht mehr zu findende Bluetooth-Geräte informiert zu werden und überlässt die Suche von Bluetooth-Geräten dem Bluetooth-Discoverer-Plugin.

Um für jedes neue lokal erreichbare Bluetooth-Gerät eine virtuelle IP-Adresse beim DHCP-Server zu erfragen, legt das Gateway-Plugin bei der Ethernet-Schnittstelle, über die der DHCP-Server zu erreichen ist, jeweils eine Dummy-Schnittstelle mit einer freien Alias-Nummer und einer Dummy-IP an und speichert die Alias- sowie die dazugehörige MAC-Adresse in einer Liste. Mit Hilfe beider Adressen wird das

Programm DHCPCD genutzt, um der Dummy-Schnittstelle eine in der Ethernet-Zelle gültige IP-Adresse zuzuweisen. Befinden sich z. B. drei mobile Bluetooth-Geräte in der Reichweite des GPAPs, so werden wie in Abbildung 4.3 dargestellt, zuerst die Dummy-Schnittstellen eth0:1, eth0:2 und eth0:3 angelegt und ihnen anschließend gültige IP-Adressen zugewiesen.

```
root@gpap:# ifconfig eth0
eth0    Link encap:Ethernet HWaddr 00:18:f3:a9:b9:13
          inet addr:10.10.129.188 Bcast:10.10.131.255 Mask:255.255.252.0
eth0:1 Link encap:Ethernet HWaddr 00:18:f3:a9:b9:13
          inet addr:10.10.128.187 Bcast:10.255.255.255 Mask:255.0.0.0
eth0:2 Link encap:Ethernet HWaddr 00:18:f3:a9:b9:13
          inet addr:10.10.130.182 Bcast:10.255.255.255 Mask:255.0.0.0
eth0:3 Link encap:Ethernet HWaddr 00:18:f3:a9:b9:13
          inet addr:10.10.129.184 Bcast:10.255.255.255 Mask:255.0.0.0
```

Abbildung 4.3 Zuordnung virtueller IP-Adressen zu virtuellen Schnittstellen

Diese IP-Adressen entsprechen nun jeweils einer virtuellen IP-Adresse eines bestimmten Bluetooth-Gerätes und werden durch einen ARP Cache Update mit der jeweiligen virtuellen IP-Adresse und der MAC-Adresse der Ethernet-Schnittstelle des GPAPs im IP-Netz bekannt gemacht. Dadurch werden alle IP-basierten Geräte darüber informiert, dass diese virtuellen IP-Adressen über den GPAP erreichbar sind. Die aktualisierte ARP-Tabelle des GPAPs ist in Abbildung 4.4 dargestellt.

```
root@gpap:# arp -a
gpap.local (10.10.128.187) at 00:18:F3:A9:B9:13 [ether] PERM on eth0
gpap.local (10.10.130.182) at 00:18:F3:A9:B9:13 [ether] PERM on eth0
gpap.local (10.10.129.184) at 00:18:F3:A9:B9:13 [ether] PERM on eth0
```

Abbildung 4.4 ARP-Tabelle nach Vergabe virtueller IP-Adressen

Um die Kommunikation zwischen IP- und nicht-IP-basierten Geräten zu ermöglichen, setzt das Gateway-Plugin die Bibliothek libnetfilter_queue [116] ein, die eine konditionale Verarbeitung und Manipulation der an einer Netzwerkschnittstelle eingehenden sowie ausgehenden Datenpakete ermöglicht, bevor bzw. nachdem der Betriebssystem-Kern sie verarbeitet. Diese Datenpakete werden in einer Liste hinterlegt und vom Gateway-Plugin abgearbeitet. Bei der Evaluation des Gateway-Plugins wurde unter anderem die Abarbeitung von ICMP Echo-Request Paketen untersucht, die in IP-Netzen typischerweise zum Austausch von Informations- und Statusmeldungen, aber auch zur Berechnung von Latenzzeiten mit dem *Internet Control Message Protocol* (ICMP) [122] eingesetzt werden. In den meisten Betriebssystemen ist diese Latenzmessung mit dem Program *ping* für IP-basierte Netze und *l2ping* für nicht-

IP-basierte Netze auf Basis von MAC-Adressen möglich. Dabei sendet das *ping*-Programm einen ICMP Echo-Request an eine angegebene IP-Adresse und wartet auf eine entsprechende Antwort in Form eines ICMP Echo-Reply Paketes. Durch Messung der Zeit für den Hin- und Rückweg der Pakete und Halbierung dieser Messzeit, ergibt sich daraus die Latenz der Verbindung zwischen beiden Kommunikationspartnern. Für die Evaluierung des Gateway-Plugins wurde das in Abbildung 4.5 dargestellte Szenario verwendet und die Programme *ping* bzw. *l2ping* zur Messung der Latenz zwischen ausgewählten Netzwerkschnittstellen eingesetzt.

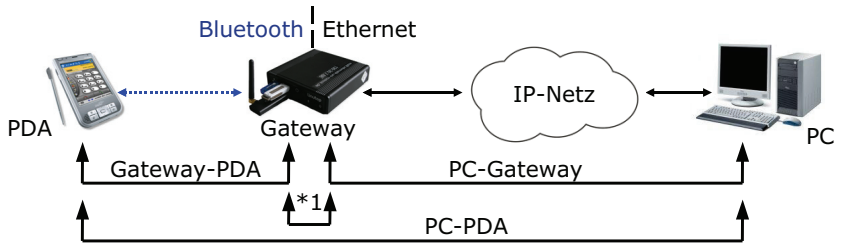

Abbildung 4.5 Szenario der Evaluation des Bluetooth-Ethernet-Gateways

Für die Messung der Latenz zwischen der IP-basierten Ethernet-Schnittstelle des PCs und der des Gateways wurde *ping* verwendet. Die Latenz der Bluetooth-Kommunikation wurde zwischen dem Gateway und dem PDA mit *l2ping* gemessen. Durch Nutzung des Bluetooth-Ethernet-Gateway-Plugins wurde es weiterhin möglich, die Latenz der Kommunikation zwischen dem PC und dem PDA zu messen, da sowohl die vom PC aus an die virtuelle IP-Adresse des Bluetooth-Gerätes gesendeten ICMP-Requests im Gateway-Plugin auf *l2ping*-ICMP-Requests an den PDA als auch die dazugehörigen *l2ping*-ICMP Echo-Reply Pakete vom PDA im Gateway-Plugin in IP-basierte ICMP Echo-Reply Pakete an den PC umgesetzt wurden. Für die Latenzmessungen wurden die Ausgaben der Programme *ping* und *l2ping* so modifiziert, dass sie eine systematische Messung mit unterschiedlichen Nutzdatengrößen der ICMP-Pakete, angefangen mit 10 Bytes bis 1480 Bytes, in 10er Schritten und einer jeweils 5-maligen Messung mit einem Abstand von 0,1 Sekunden ermöglichten. Weiterhin speichern sie die Messdaten in Textdateien, sodass eine anschließende Analyse mit Statistik-Werkzeugen erfolgen konnte. Auf diese Weise wurden die in Abbildung 4.6 dargestellten Messreihen mit ihren dazugehörigen Regressionsgeraden für die in Abbildung 4.5 dargestellten Messszenarien ermittelt.

Die in der Grafik mit *PC-Gateway* bezeichnete Messreihe stellt die Latenz der IP-basierten Ethernet-Verbindung zwischen dem PC und dem Gateway bei Verwendung ansteigender Nutzdatengrößen der Pakete dar. Die darüberliegende Messreihe *Gateway-PDA* zeigt das Latenzverhalten der Bluetooth-Verbindung zwischen dem

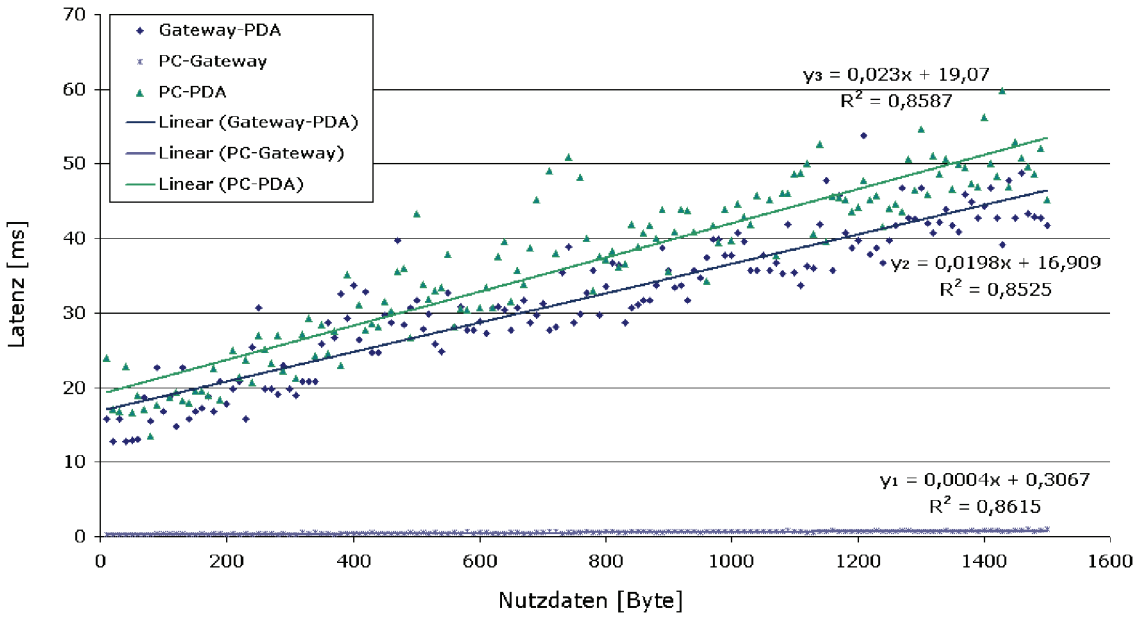

Abbildung 4.6 Latenzmessungen mit dem Bluetooth-Ethernet-Gateway-Plugin

Gateway und dem PDA. Hierbei ist erkennbar, dass der Austausch von Datenpaketen in der Bluetooth-Zelle nicht nur deutlich länger dauert, sondern mit zunehmender Paketlänge stärker ansteigt. Die Werte der Messreihe *PC-PDA* liegen noch etwas höher und steigen mit zunehmender Paketlänge noch etwas stärker an. Bevor genauer auf die Bedeutung der Verläufe der ermittelten Messreihen und den Einfluss des Gateway-Plugins auf die Kommunikation eingegangen wird, soll an dieser Stelle mit Hilfe der Abbildung 4.7 kurz der theoretische Zusammenhang zwischen Latenz und Bandbreite einer Kommunikationsverbindung erläutert werden.

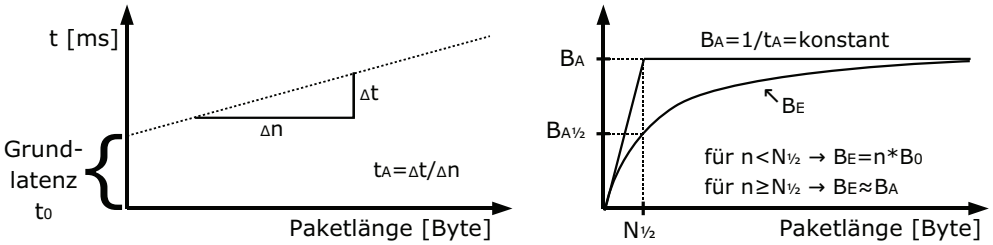

Abbildung 4.7 Zusammenhänge bei der Bestimmung der Bandbreiten

Im linken Diagramm der Abbildung 4.7 ist der Zusammenhang zwischen der Paketlänge und der Latenz eines Datenpaketes dargestellt. Die Gerade im Diagramm stellt eine Regressionsgerade dar, die durch einen Anfangswert und einen Anstieg eindeutig definiert wird. Der Anfangswert t_0 entspricht der Grundlatenz, die beim Routing

eines Paketes ohne Nutzdaten entsteht. Der Anstieg der Geraden stellt die Versandzeit je Byte dar und wird t_A genannt. Die konstante maximale Bandbreite B_A ergibt sich dann aus der Division von 1 und t_A. Der Zusammenhang der Paketgröße und der effektiven Bandbreite B_E ist im rechten Diagramm dargestellt. Ist die Paketlänge gleich 0, so ist auch die effektive Bandbreite gleich 0. Wird die Paketlänge vergrößert, steigt die effektive Bandbreite an und konvergiert mit großen Paketen gegen die maximale Bandbreite B_A. Um die effektive Bandbreite besser beschreiben zu können, wird nach [97] die Paketgröße, bei der die effektive Bandbreite den Wert $B_{A1/2}$ erreicht, $N_{1/2}$ genannt. Für Paketlängen kleiner als $N_{1/2}$ wird die effektive Bandbreite näherungsweise durch eine Gerade mit dem Anstieg B_0 beschrieben. Für Pakete mit einer größeren Länge wird die effektive Bandbreite durch B_A angenähert.

Die Anwendung des Zusammenhangs von Latenz und Bandbreite auf die Messreihen in Abbildung 4.6 macht deutlich, dass die Grundlatenz der Übertragung von Bluetooth-Paketen deutlich höher ist als die für Ethernet-Pakete. Dieser Sachverhalt ist nicht wirklich überraschend, wenn man die verwendeten Übertragungsmedien beider Netzwerktechnologien bedenkt. Weiterhin lässt sich mit den Messreihen die zusätzliche Grundlatenz durch den Einsatz der Gateway-Funktionalität auf dem GPAP bestimmen, indem von der Grundlatenz der Messreihe *PC-PDA* die Grundlatenzen der Messreihen *PC-Gateway* und *Gateway-PDA* subtrahiert werden. Hierbei ergibt sich eine Grundlatenz der Gateway-Funktionalität von $t_{Gateway}$ = (19,07-16,91-0,31) ms = 1,85 ms für die gesamte Hin- und Rückrichtung. Das bedeutet, dass die Gateway-Funktionalität des GPAPs für die Vermittlung von Datenpaketen zwischen Bluetooth- und Ethernet-basierten Netzwerken für eine Grundlatenz von t = 0,925 ms verantwortlich ist, was beim Vergleich mit der Latenz der reinen Bluetooth-Kommunikation vertretbar erscheint.

Mit Hilfe der Regressionsgeraden in Abbildung 4.6 können weiterhin die theoretischen maximalen Bandbreiten und der Einfluss des GPAPs auf die Bandbreite zwischen dem Ethernet-basierten PC und dem Bluetooth-basierten PDA berechnet werden. Dabei wird auf die in Abbildung 4.7 dargestellten Zusammenhänge zurückgegriffen.

Bei den zur linearen Regression verwendeten Latenzmessungen ist zu beachten, dass die Zeiten für den Hin- und Rückweg der Pakete gemessen wurden. Aus diesem Grund müssen die Anstiege sowie die Grundlatenzen vor der Berechnung halbiert werden. Für die theoretische maximale Bandbreite der Ethernet-Verbindung in Messreihe *PC-Gateway* ergibt sich damit die folgende Berechnung:

$$2 * t = t_A * p + t_0 \tag{4.1}$$

$$2 * t = 0,0004\frac{ms}{Bytes} * p + 0,3067ms \tag{4.2}$$

$$t = 0,0002\frac{ms}{Bytes} * p + 0,15335ms \tag{4.3}$$

$$t_A = 0,0002\frac{ms}{Bytes} \tag{4.4}$$

$$B_A = \frac{1}{t_A} = \frac{1}{0,0002\frac{ms}{Bytes}} = \frac{1}{0,0000002\frac{s}{Bytes}} = 4882,81\frac{KB}{s} \tag{4.5}$$

Der aus der Grafik ablesbare Wert für t_A hat vier signifikante Stellen. Die 5. Stelle ist unsicher und gibt damit die Messunsicherheit mit $\triangle t = 0{,}00005$ an. Der absolute Fehler ergibt sich aus der folgenden Berechnung:

$$2 * t = 0,0004\frac{ms}{Bytes} * p + 0,3067ms \tag{4.6}$$

$$t_A = 0,0004\frac{ms}{Bytes} \tag{4.7}$$

$$t_{absolut} = 0,0004\frac{ms}{Bytes} \pm 0,00005\frac{ms}{Bytes} \tag{4.8}$$

Daraus resultiert der folgende relative Fehler:

$$t_{relativ} = 0,0004\frac{ms}{Bytes} \pm \frac{0,00005\frac{ms}{Bytes}}{0,0004\frac{ms}{Bytes}} * 100\% \tag{4.9}$$

$$t_{relativ} = 0,0004\frac{ms}{Bytes} \pm 12,5\% \tag{4.10}$$

Die theoretische maximale Bandbreite der Bluetooth-Verbindung zwischen dem Gateway und dem PDA, deren Latenz in der Messreihe *Gateway-PDA* gemessen wurde, berechnet sich in gleicher Weise:

$$2 * t = t_A * p + t_0 \tag{4.11}$$

$$2 * t = 0,0198 \frac{ms}{Bytes} * p + 16,909 ms \tag{4.12}$$

$$t = 0,0099 \frac{ms}{Bytes} * p + 8,4545 ms \tag{4.13}$$

$$t_A = 0,0099 \frac{ms}{Bytes} \tag{4.14}$$

$$B_A = \frac{1}{t_A} = \frac{1}{0,0099 \frac{ms}{Bytes}} = \frac{1}{0,0000099 \frac{s}{Bytes}} = 98,64 \frac{KB}{s} \tag{4.15}$$

Der aus der Grafik ablesbare Wert für t_A hat auch hier vier signifikante Stellen. Die 5. Stelle ist wieder unsicher und gibt damit die Messunsicherheit mit $\triangle t = 0{,}00005$ an. Daraus ergeben sich folgende absolute und relative Fehler:

$$t_A = 0,0198 \frac{ms}{Bytes} \tag{4.16}$$

$$t_{absolut} = 0,0198 \frac{ms}{Bytes} \pm 0,00005 \frac{ms}{Bytes} \tag{4.17}$$

$$t_{relativ} = 0,0198 \frac{ms}{Bytes} \pm \frac{0,00005 \frac{ms}{Bytes}}{0,0198 \frac{ms}{Bytes}} * 100\% \tag{4.18}$$

$$t_{relativ} = 0,0198 \frac{ms}{Bytes} \pm 2,53\% \tag{4.19}$$

Zu guter letzt wird nun die theoretische maximale Bandbreite der Verbindung für die Kommunikation zwischen dem Ethernet-basierten PC und dem Bluetooth-basierten PDA unter Nutzung des Gateways als Vermittler zwischen beiden Netzwerktechnologien anhand der Messreihe *PC-PDA* bestimmt:

$$2 * t = t_A * p + t_0 \tag{4.20}$$

$$2 * t = 0,0230\frac{ms}{Bytes} * p + 19,070ms \tag{4.21}$$

$$t = 0,0115\frac{ms}{Bytes} * p + 9,535ms \tag{4.22}$$

$$t_A = 0,0115\frac{ms}{Bytes} \tag{4.23}$$

$$B_A = \frac{1}{t_A} = \frac{1}{0,0115\frac{ms}{Bytes}} = \frac{1}{0,0000115\frac{s}{Bytes}} = 84,92\frac{KB}{s} \tag{4.24}$$

Auch bei dieser Messreihe ist die 5. Stelle unsicher und gibt damit die Messunsicherheit mit $\triangle t = 0{,}00005$ an. Der absolute und der relative Fehler lassen sich wie folgt berechnen:

$$t_A = 0,0230\frac{ms}{Bytes} \tag{4.25}$$

$$t_{absolut} = 0,0230\frac{ms}{Bytes} \pm 0,00005\frac{ms}{Bytes} \tag{4.26}$$

$$t_{relativ} = 0,0230\frac{ms}{Bytes} \pm \frac{0,00005\frac{ms}{Bytes}}{0,0230\frac{ms}{Bytes}} * 100\% \tag{4.27}$$

$$t_{relativ} = 0,0230\frac{ms}{Bytes} \pm 2,17\% \tag{4.28}$$

Zusammenfassend zeigen die ermittelten Messreihen, dass das Konzept der heterogenen Adressierung zwischen der IP-basierten Ethernet-Zelle und der nicht-IP-basierten Bluetooth-Zelle am Beispiel der Vermittlung von ICMP-Paketen durch den GPAP funktioniert und er mit einem Wert von t = 0,925 ms nur einen geringen Einfluss auf die zusätzliche Grundlatenz der Kommunikation hat. Der leicht höhere Anstieg der Regressionsgerade wirkt sich auf die für eine heterogene Kommunikation zur Verfügung stehende maximale Bandbreite aus, die in diesem Fall von 98,64 KB/s auf 84,92 KB/s verringert wird. Damit liegt der Einfluss des GPAPs auf die Latenz und die Bandbreite etwa in dem Bereich, den auch ein zusätzlicher Switch oder Hub auf dem Kommunikationspfad im IP-Netz verursachen würde [123]. Der ungewöhnlich hohe relative Fehler der Messung der Ethernet-Verbindung deutet darauf hin, dass die mit dem *ping*-Programm ermittelten Latenzzeiten mit zu wenig

Nachkommastellen berechnet und ausgegeben werden. Für die Latenzmessungen, die Bluetooth-Verbindungen enthalten, sind die verwendeten Messprogramme *ping* und *l2ping* jedoch hinreichend genau.

4.3 Evaluationsszenarien des Service Proxyings

Nachdem in den vorherigen Abschnitten genauer auf die realisierten Komponenten der Netzwerkschicht des GPAPs und die Evaluation des Konzepts der heterogenen Adressierung in dynamischen Ensembles eingegangen wurde, stellen die nächsten Abschnitte die Evaluationsergebnisse für drei unterschiedliche Szenarien des in Abschnitt 3.5.1 vorgestellten Konzepts des Service Proxyings detaillierter vor.

4.3.1 Nachrichten- und Dateiaustausch zwischen IP-basierten Web Services und Bluetooth-SDP-Diensten

Das erste Evaluationsszenario beschäftigt sich mit dem Nachrichten- und Dateiaustausch zwischen Endgeräten, die entweder über eine Ethernet- oder eine Bluetooth-Schnittstelle verfügen und somit unterschiedliche Netzwerktechnologien zur Kommunikation einsetzen. Mit der Nutzung verschiedener Netzwerktechnologien geht in diesem Szenario wieder die Nutzung unterschiedlicher Service-Technologien einher. Für den Datenaustausch wird dabei auf das in Abbildung 3.16 dargestellte Szenario des Service Proxyings und die Nutzung von Web Services in der Ethernet-Zelle sowie SDP-Diensten in der Bluetooth-Zelle zurückgegriffen [124].

Für die Evaluation des Nachrichten- und Dateiaustauschs wurden zunächst prototypische Java-Anwendungen entwickelt, die den Datenaustausch innerhalb ihrer homogenen Zelle ohne die Nutzung eines Proxys realisieren. Dazu wurde zuerst eine grafische Oberfläche (engl. Graphical User Interface (GUI)) für die mobilen Geräte in der Bluetooth-Zelle entwickelt, die es Bluetooth-Geräten ermöglicht, einen SDP-Dienst zur Signalisierung der eigenen Bereitschaft zur Datenannahme in der Rolle eines SDP-Providers zu veröffentlichen und Daten von anderen Bluetooth-Geräten, die diesen Dienst nutzen möchten, entgegenzunehmen. Weiterhin ermöglicht die GUI eine Suche nach benachbarten Bluetooth-Geräten und deren SDP-Diensten. Dabei werden die Geräte, die einen Dienst zum Nachrichten- und Dateiaustausch unter einer speziellen UUID anbieten, in einer Teilnehmerliste der GUI gespeichert. Aus dieser Liste kann dann ein Teilnehmer ausgewählt werden und eine Nachricht oder eine Datei an ihn übertragen werden. Für die Datenübertragung werden in der Bluetooth-Zelle RFCOMM-Verbindungen und als Format für die zu übermittelnden Daten das *Tag Length Value* (TLV)-Format verwendet. Hierbei bestimmt das erste Byte die

Art der Daten (Nachricht, oder Datei) und die nächsten zwei Bytes die Länge der anschließenden Nutzdaten, die je nach Art der Daten beim Empfänger verarbeitet werden. Für die Bereitstellung von SDP-Diensten sowie die Suche nach benachbarten Bluetooth-Geräten und deren SDP-Diensten, wurde auf die Bluetooth-Bibliothek *avetana* [125] zurückgegriffen, die für die Programmiersprache Java frei verfügbar ist.

Auch für die Realisierung des Nachrichten- und Dateiaustauschs in der Ethernet-Zelle wurde eine prototypische GUI entwickelt. Diese nutzt jedoch Web Services zum Datenaustausch. Einerseits bietet die GUI die Möglichkeit, die Bereitschaft der Datennannahme in der Rolle eines WS-Providers als Web Service mit den Parametern IP, Port und Pfad in einem WSIL-Verzeichnis beim Broker zu veröffentlichen (vgl. Abschnitt 2.2.4). Andererseits ermöglicht die GUI, das WSIL-Verzeichnis nach Diensten zu durchsuchen und an Informationen wie z. B. den Pfad, den Namen, eine optionale Kurzbeschreibung sowie die Adresse der WSDL-Beschreibung eines passenden Dienstes zu gelangen. Dabei wird zum Auswerten der XML-Datei des WSIL-Verzeichnisses die Java-Bibliothek JDOM [126] herangezogen, die Parser-Funktionalitäten für XML-Dokumente bereitstellt. Bei einer ersten Nutzung eines Dienstes ist es für den Consumer einmalig erforderlich, aus der WSDL-Beschreibung mit Hilfe der Anwendung *wsimport* die Java-Schnittstellenklassen eines Dienstes zu generieren. Dabei werden aus den beschriebenen Operationen des Web Services Java-Methoden erzeugt, welche die GUI in der Rolle des Consumers für die Dienstnutzung verwendet. Ohne diese Schnittstellenklassen kann der Web Service nicht genutzt werden. Anschließend werden die gefundenen Dienste in die Teilnehmerliste der GUI eingetragen. Nach Eingabe einer Nachricht bzw. nach Auswahl einer Datei über einen Auswahldialog, werden sie als Objekt in SOAP gekapselt und vom Consumer zum Provider übertragen.

Nach der Implementierung des Dienst-basierten Nachrichten- und Dateiaustauschs für beide Service-Technologien, wurde ein Service-Proxy realisiert, der die Funktionalitäten des in Abschnitt 3.5.1 beschriebenen Service Proxyings am Beispiel von Web Services und Bluetooth-SDP-Diensten umsetzt. Der Service-Proxy realisiert sowohl die Suche nach Web Services in der Ethernet-Zelle und die Bereitstellung äquivalenter SDP-Dienste in der Bluetooth-Zelle, als auch die entgegengesetzte Proxy-Richtung. Dabei wird für jeden zu vermittelnden Dienst ein Thread gestartet, der auf eingehende Verbindungen wartet und die Daten an den äquivalenten Dienst weitervermittelt. Die Funktionalität der Web Services und SDP-Dienste wurde dabei nicht verändert.

Bei der Evaluation des Service Proxyings wurden die in Abschnitt 3.5.1 beschriebenen Kommunikationsszenarien untersucht, die a) den direkten SDP-basierten Datenaustausch innerhalb einer Bluetooth-Zelle; b) den direkten WS-basierten Datenaus-

tausch in einer Ethernet-Zelle und c) den Datenaustausch zwischen einer Ethernet-Zelle und einer Bluetooth-Zelle unter Einsatz des Service Proxyings zwischen Web Services und SDP-Diensten sowie den zweifachen Einsatz von Proxys für den Datenaustausch zwischen entfernten Bluetooth-Zellen evaluieren.

a) Evaluation der Dienstnutzung innerhalb von Bluetooth-Zellen
Als Einflussfaktoren auf die Datenübertragung wurde sowohl die Größe der zu übertragenen Datei als auch die Payload-Größe der dabei verwendeten Datenpakete betrachtet. Der Einfluss der Dateigröße ist in Abbildung 4.8 für einen SDP-basierten Datenaustausch zwischen mobilen Bluetooth-Geräten dargestellt, wobei Dateien der Größe 20 KB, 200 KB und 2000 KB mit konstanter Payload-Größe von 1000 Bytes jeweils mehrmals übertragen und anschließend die durchschnittliche Übertragungszeit ermittelt wurde.

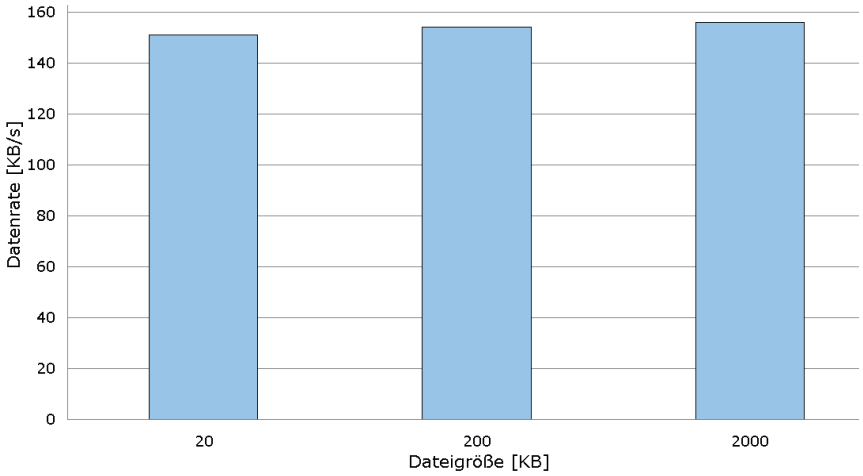

Abbildung 4.8 SDP-SDP-Kommunikation mit variierter Dateigröße

Die kleine Datei wurde innerhalb von 0,14 s, die mittelgroße in 1,3 s und die große Datei in 12,91 s übertragen. Da die verwendete Dateigröße kaum einen Einfluss auf die erzielbare Datenrate von etwa 160 KB/s hat, wurde für die nächsten Messungen nur noch die mittelgroße Datei mit einer Größe von 200 KB eingesetzt und dafür die Payload-Größe variiert. Die Ergebnisse dieser Messung innerhalb einer Bluetooth-Zelle sind in Abbildung 4.9 dargestellt.

Diese Messungen zeigen, dass die Variation der Payload-Größe ab einer Größe von 10 Bytes aufwärts kaum einen Einfluss auf die erzielte Datenrate der Übertragung hat. Nur bei einer Payload-Größe von unter 10 Bytes je Datenpaket wirkte sich das schlechte Verhältnis von Header und Payload der Bluetooth-Pakete negativ auf die erzielbare Datenrate aus.

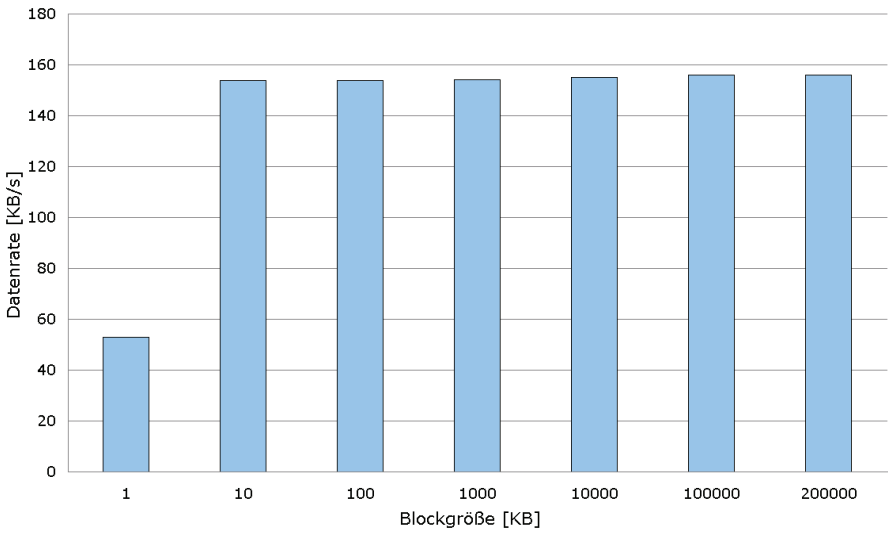

Abbildung 4.9 SDP-SDP-Kommunikation mit variierter Blockgröße

b) Evaluation der Dienstnutzung innerhalb von Ethernet-Zellen
Bei der Nutzung von Web Services innerhalb einer Ethernet-Zelle konnte durch Variation der Payload-Größe ein viel stärkerer Einfluss auf die Kommunikation eruiert werden.

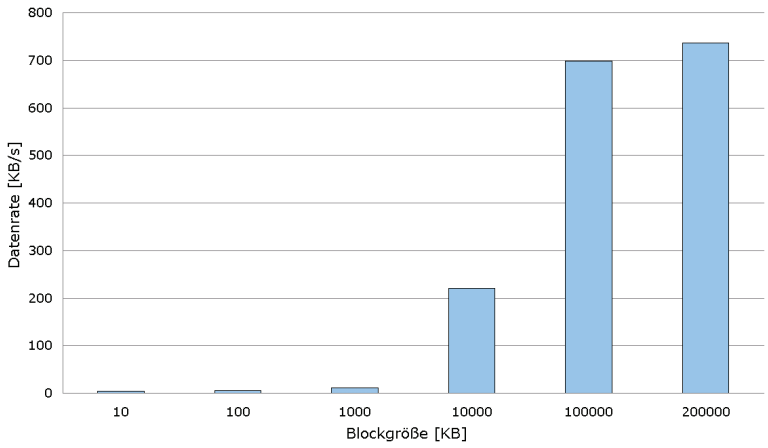

Abbildung 4.10 WS-WS-Kommunikation mit variierter Blockgröße

Die in Abbildung 4.10 dargestellten Datenraten zeigen, dass sich für die Nutzung von Web Services und des dabei verwendeten SOAP-Protokolls erst ab einer Payload-Größe von 100.000 Bytes eine etwa gleichbleibende Datenrate einstellt. Hierbei handelt es sich nicht um die Größe eines einzelnen Datenpaketes, sondern um die Größe einer SOAP-Nachricht, die aufgeteilt auf mehrere TCP-Pakete übertragen wird. Dies deutet darauf hin, dass der Einsatz von Web Services und SOAP zu einem hohen

Overhead während der Kommunikation führt. Gerade für mobile Geräte, die im Rahmen des MuSAMA-Projekts vor allem kurze Kontextinformationen mit benachbarten Geräten innerhalb eines Ensembles austauschen, stellt dies einen gravierenden Nachteil von Web Services dar. Da sich durch diese Messungen eine Payload-Größe im Bereich von 100.000 Bytes als halbwegs praktikabel für die Nutzung von Web Services herausgestellt hat, wird dieser Wert in den folgenden Evaluationen des Service Proxyings eingesetzt.

c) Evaluation des Service Proxyings zwischen Bluetooth- und Ethernet-Zellen sowie deren Kombination
Bei der Evaluation des einmaligen Service Proxyings zur Vermittlung zwischen Web Services der Ethernet-Zelle und SDP-Diensten der Bluetooth-Zelle wurde eine Datenrate von 145,5 KB/s erreicht. Für das Szenario der Kommunikation zwischen entfernten Bluetooth-Zellen wurde das Konzept des Service Proxyings mit zwei Proxys realisiert. Die dabei ermittelte Datenrate beträgt 111,4 KB/s und zeigt, dass der zweimalige Proxy-Einsatz die erreichbare Datenrate zusätzlich verringert.

In Abbildung 4.11 werden die in den Evaluationsszenarien ermittelten Datenraten noch einmal gegenübergestellt. Die grünen Balken stellen dabei die Datenraten dar, die aus Sicht der zu übertragenen Datei erreicht werden konnten. Die blauen Balken geben die bei den jeweiligen Szenarien ebenfalls gemessenen Datenraten aus Sicht der Bluetooth-Schnittstelle an, die den Overhead der Paket-Header mit enthalten. Die schwarzen Balken stellen die verwendete Datenrate an den Ethernet-Schnittstellen dar und enthalten den Overhead der Ethernet-, IP-, TCP-, HTTP- und SOAP-Header, die während der WS-WS-Kommunikation benötigt werden.

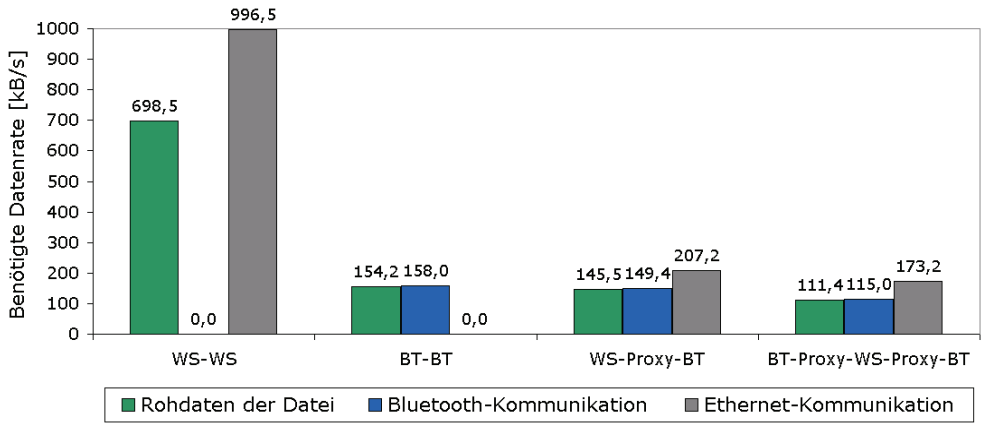

Abbildung 4.11 Vergleich der erzielten Datenraten und des dabei ermittelten Kommunikations-Overheads [127]

Bei den in Abbildung 4.11 gegenübergestellten Messungen wird vor allem der Overhead der auf dem SOAP-Protokoll basierten WS-WS-Kommunikation ersichtlich, der sich bei der Datenübertragung mit einem Kommunikations-Overhead von 42,6% deutlich auf die real benötigte Datenrate der Netzwerkschnittstelle niederschlägt. Der Einsatz von SDP in Bluetooth-Zellen hat dagegen mit einem Overhead von 2,5% kaum eine Auswirkung auf die in der Anwendungsebene nutzbare Datenrate. Ebenfalls hat der Einsatz der Proxys nur eine geringe Auswirkung auf die während der Service-basierten Kommunikation erzielbare Datenrate.

4.3.2 Service-basierte Individualkommunikation zwischen Teilnehmern von virtuellen und Präsenzlehrveranstaltungen

Die rechnergestützte Präsenzlehre gehört derzeit zu den am weitesten verbreiteten E-Learning-Formen. Sie beinhaltet den Austausch von Vortragsfolien und die Ergänzung von Tafelbildern durch digitale Medien. Die Prozesse der Präsenzlehre werden bei der virtuellen Lehre nachgebildet und oft um spezifische Elemente der Kommunikation und Kollaboration, wie z. B. einem Chat, einem Forum oder einem Shared Whiteboard angereichert. Die Kombination beider Lernparadigmen wurde am Lehrstuhl für Rechnerarchitektur im Rahmen des MuSAMA-Projekts durch das in Abbildung 4.12 dargestellte Szenario einer Dienst-basierten Kopplung von virtueller und Präsenzlehre realisiert [128].

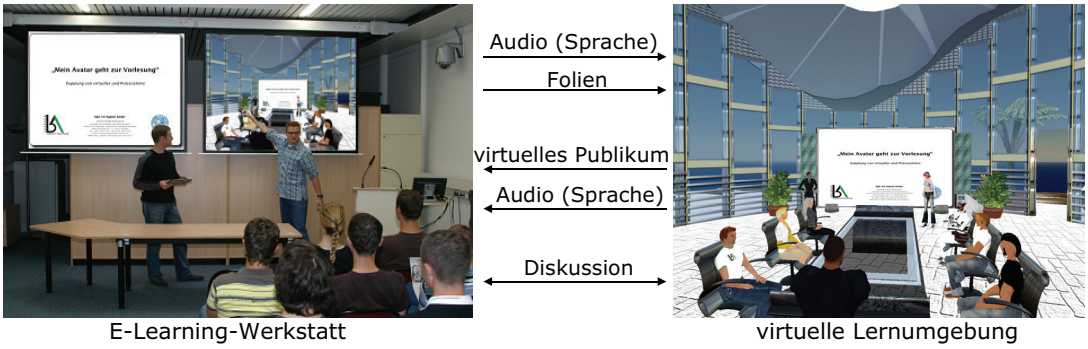

Abbildung 4.12 Service-basierte Kopplung von virtueller und Präsenzlehre [128]

Als reale Lehr- und Lernumgebung wurde die E-Learning-Werkstatt des Lehrstuhls für Rechnerarchitektur eingesetzt, die über verschiedene Schnittstellen zur Nutzung und Steuerung von Kameras, Bildschirmen, Projektoren, Mikrofonen und Lautsprechern sowie weitere Anschlussmöglichkeiten für Laptops, PCs und zusätzliche Multimedia-Technik verfügt. Als virtuelle Lehr- und Lernumgebung wurde ein virtueller Vorlesungsraum auf der Parzelle des *European University Island* in der virtuellen

Welt *Second Life* (SL) [129] entwickelt, der eine offene Kommunikationsplattform bereitstellt. Mit der aktuellen Ausstattung der E-Learning-Werkstatt stehen die technischen Grundlagen für das Streamen von Video- und Audiodaten in die virtuelle Umgebung Second Life bereit. Auf diese Weise können sowohl die Vortragsfolien als auch die Sprache des Vortragenden von der realen in die virtuelle Umgebung übertragen und die Folien in Form eines Videostreams auf einer virtuellen Leinwand dargestellt werden. Gleichzeitig ist der Vortragende mit seinem Avatar im SL vertreten, über den virtuelle Zuhörer die Sprache des realen Vortragenden wahrnehmen. In umgekehrter Richtung nutzt der Vortragende in der realen Umgebung einen SL-Client auf seinem Laptop, um das virtuelle Publikum zu sehen und mit ihm Diskutieren zu können. Auf diese Weise wird es Lernenden ermöglicht, nicht nur in der realen Umgebung der E-Learning-Werkstatt sondern auch in der virtuellen Umgebung Second Life an Lehrveranstaltungen teilnehmen zu können.

Da die kollaborative Vor- und Nachbereitung für den individuellen Lernprozess unabdingbar ist, wurde das beschriebene Lehr- und Lernparadigma um eine in Abbildung 4.13 dargestellte Service-basierte Individualkommunikation zwischen virtuell und real lernenden Teilnehmern erweitert [130].

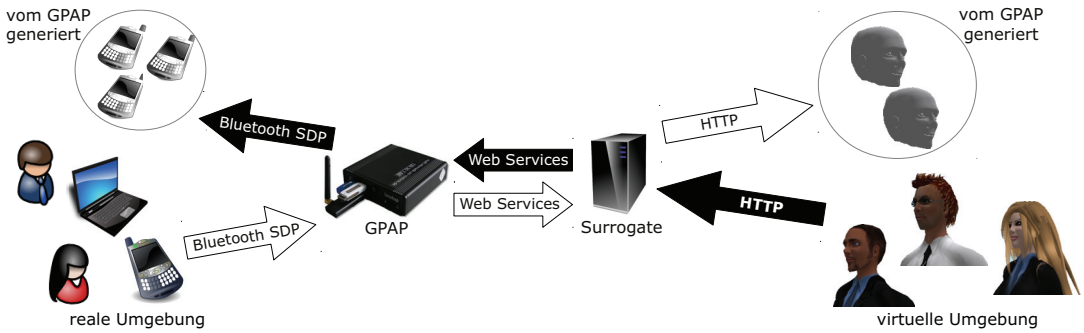

Abbildung 4.13 Service-basierte Individualkommunikation zwischen realen und virtuellen Umgebungen [130]

Während virtuelle Teilnehmer ihren Avatar mit einem realisierten virtuellen Headset ausstatten können, nutzen reale Teilnehmer einen Chat-Client auf ihrem mitgebrachten und mit Bluetooth ausgestatteten Mobiltelefon, PDA oder Laptop zur Kommunikation. Die dazu notwendigen grafischen Nutzungsoberflächen sind in Abbildung 4.14 dargestellt. Dabei handelt es sich im linken Teil des Bildes um die Hauptmenüs zum Lesen und Schreiben von Nachrichten der JavaME-basierten GUI, die für Mobiltelefone unter Zuhilfenahme des Simulators des *Wireless Toolkits* (WTKs) der Firma *Sun Microsystems* entwickelt und anschließend auf Mobiltelefonen vom Typ Nokia 6230i und Nokia 6300 evaluiert wurde. Im rechten Teil des Bildes ist dagegen die Chat-Komponente im Second Life dargestellt.

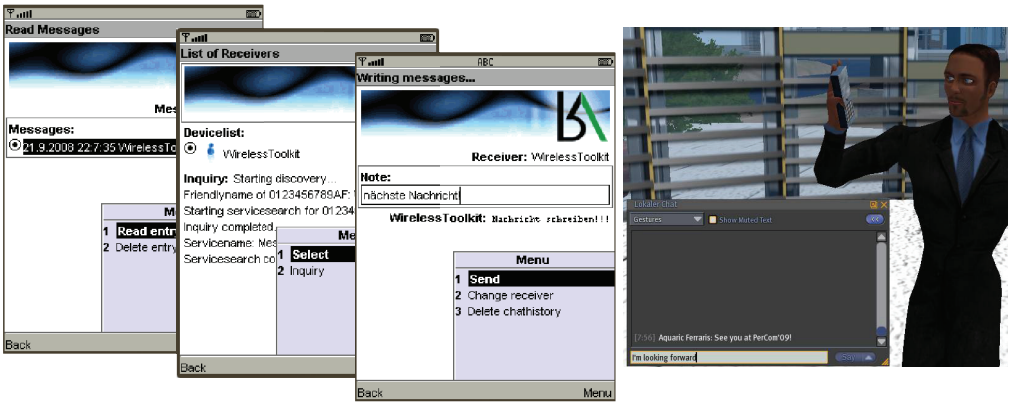

Abbildung 4.14 Chat-Komponenten in der realen und virtuellen Umgebung [130]

Auf den mobilen Bluetooth-basierten Geräten werden wie in den vorherigen Abschnitten beschrieben, SDP-Dienste zur Bereitstellung, Suche und Nutzung der Nachrichtenübertragung eingesetzt. Für die Chat-Anwendung der virtuellen Teilnehmer werden dagegen Web Services zur Kommunikation genutzt. Da Second Life zwar die Nutzung von HTTP-Requests ermöglicht, jedoch Web Services noch nicht direkt nutzen kann, musste ein Surrogate entwickelt werden, das zwischen HTTP-Requests und Web Services vermittelt. Der GPAP wurde wie im vorherigen Evaluationsszenario für das in Abschnitt 3.5.1 beschriebene Service Proxying zwischen Bluetooth-SDP-Diensten und Web Services eingesetzt. Damit ergeben sich auch für die Chat-Funktionalität zwischen realen und virtuellen Teilnehmern mehrere Varianten der Kommunikation.

Für die Kommunikation zwischen benachbarten Bluetooth-Geräten erzeugen deren Chat-Clients jeweils einen SDP-Dienst, der von anderen Geräten gefunden und genutzt werden kann. Die Möglichkeit des direkten Bluetooth-basierten Nachrichtenaustauschs ist also auch ohne den GPAP und das Surrogate gegeben.

Um Nachrichten zwischen den Avataren in Second Life austauschen zu können, legen Avatare ihr virtuelles Headset an, das sie beim Anlegen am Surrogate registriert. Das Surrogate erzeugt dann jeweils einen Web Service, der unter anderem den Namen des Avatars enthält und trägt ihn im WSIL-Verzeichnis des Brokers ein. Anschließend werden die aktuellen Web Services für die bereits registrierten Avatare sichtbar gemacht. Nach Auswahl eines Kommunikationsteilnehmers wird die Nachricht als HTTP-Nachricht an das Surrogate gesendet und dort durch Aufruf des zugehörigen Web Services an den Teilnehmer übermittelt.

Die Kombination beider Kommunikationsformen führte zum Einsatz des GPAPs, der sowohl in der Bluetooth-Zelle nach entsprechenden SDP-Diensten sucht und für sie

äquivalente Web Services im IP-basierten Netzwerk bereitstellt, als auch im WSIL-Verzeichnis regelmäßig nach vorhandenen Web Services nachschaut und dazu äquivalente SDP-Dienste in der Bluetooth-Zelle bereitstellt.

Mit der Evaluation des Szenarios der Service-basierten Individualkommunikation zwischen Teilnehmern von virtuellen und Präsenzlehrveranstaltungen konnte der Ansatz des in Abschnitt 3.5.1 beschriebenen Service Proxyings in einem weiteren Szenario eingesetzt und näher untersucht werden. Dabei wurde gezeigt, dass der Ansatz des Service Proxyings auch hier eine transparente Kommunikation zwischen den Kommunikationspartnern ermöglicht und es für sie keinen Unterschied macht, ob sie mit einem Teilnehmer Daten austauschen, der die gleiche Service-Technologie einsetzt oder eine andere, da Dienste jedem Teilnehmer in seiner eigenen Service-Technologie zur Verfügung gestellt werden. Weiterhin konnte mit diesem Szenario auch gezeigt werden, dass es auf eine transparente Weise möglich ist, Funktionen zur Interoperabilität von Diensten vom GPAP auf z. B. ein Surrogate oder einen beliebigen Server im Internet auszulagern und als Web Service einzubinden. Dadurch ist es nicht mehr nötig, den GPAP für jedes neue Szenario um ein Plugin zur Datenvermittlung zu erweitern. Stattdessen können generelle Plugins für die Konvertierung von SDP- und Web Services eingesetzt und spezielle Funktionen wie die Vermittlung zwischen z. B. Web Services und HTTP-Requests ausgelagert werden.

4.3.3 Kontext-orientierte SOA-Interoperabilität für Broadcast-Szenarien

Ein weiteres Evaluationsszenario beschäftigte sich mit dem Einsatz des Service Proxyings für eine kontextorientierte Jukebox, die im Sinne der Adaption intelligenter Umgebungen an die Bedürfnisse der Nutzer, je nach Musikgeschmack der sie umgebenden Personen einen Webradio-Sender auswählt und abspielt, der den meisten gefallen sollte [131]. Ihre Aufgabe ist also die bestmögliche musikalische Unterhaltung der Anwesenden in z. B. einem Supermarkt, einem Wartezimmer oder einer Flughafenlounge. Eine grobe Komponentenübersicht der Lösung ist in Abbildung 4.15 dargestellt.

Der Musikgeschmack der anwesenden Personen wird mit Hilfe der in Abbildung 4.16 dargestellten GUI als Geschmacksprofil auf ihren Bluetooth-fähigen Mobiltelefonen zusammengestellt, als XML-Datei gespeichert und als SDP-Dienst veröffentlicht. Die Jukebox kann diese SDP-Dienste nicht direkt nutzen, da sie einen reinen Web Service Consumer darstellt. An dieser Stelle kommt wieder der GPAP zum Einsatz, um die als SDP-Dienst angebotenen Geschmacksprofile der Nutzer als Web Services zur Verfügung zu stellen. Für den Zugriff auf die Bluetooth-Netzwerkschnittstelle kam dabei die Java-Bibliothek *Bluecove* [132] zum Einsatz, die Funktionen für SDP,

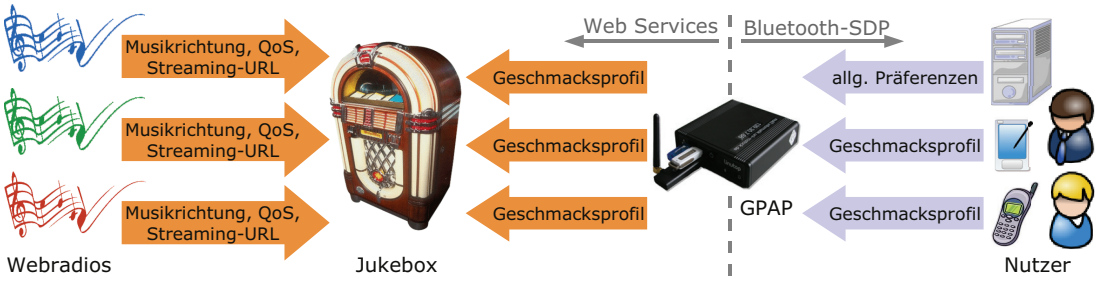

Abbildung 4.15 Komponentenübersicht des Jukebox-Systems [131]

RFCOMM, L2CAP und OBEX bereitstellt. Auch die Webradio-Sender stehen der Jukebox in Form von Web Services zur Verfügung. Bei der Auswahl, welcher Sender gespielt werden soll, wird neben der Summe der Geschmacksprofile der Nutzer, des Orts der Jukebox und der Geschwindigkeit der Internetanbindung aber auch der Codec (z. B. MP3, AAC+, OGG), die Bitrate des Radiostreams (z. B. 28 kbit/s, 56 kbit/s, 128 kbit/s) und das Musikgenre (z. B. Klassik, Rock, Schlager, Techno) berücksichtigt. Für das Abspielen der Musik wurde in der Jukebox die *JVLC*-Bibliothek genutzt, die ein Java Binding für die C++-Bibliotheken des *VLC media Players* [133] für zahlreiche Audio- und Video-Formate unterstützt.

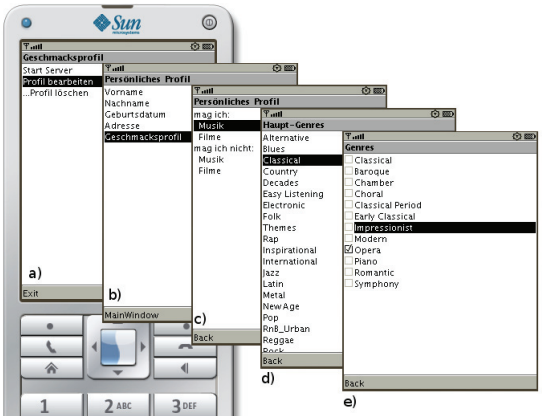

Abbildung 4.16 Screenshots einiger Menüs im Simulator [131]

Während der Evaluation der Jukebox hat sich sowohl für das Bluetooth-Discovery des GPAPs nach mobilen Geräten der Nutzer als auch für die WS-Suche beim Broker ein zeitlicher Abstand von 30 Sekunden als sinnvoll herausgestellt. Ein erkennbarer Einfluss der Größe der zwischen den mobilen Geräten und der Jukebox zu übertragenen Geschmacksprofile konnte nicht festgestellt werden, jedoch nimmt die Dienstsuche im Bluetooth-Netz mit zunehmender Geräte-Anzahl zu und BT-Geräte werden nicht bei jeder Suche gefunden, obwohl sie sich noch in Reichweite befinden.

4.4 Zusammenfassung der Evaluation

Am Anfang dieses Kapitels wurde die Hardware- und Software-Umgebung des General Purpose Access Points vorgestellt. Danach folgte die Evaluation der Realisierung der in Abschnitt 3.7 beschriebenen Referenzarchitektur des GPAPs. Dabei konnte am Beispiel des Network/Service Visualizers gezeigt werden, dass die Zusammenarbeit des HCBR-Cores mit den Device- und HCBR-Plugins problemlos funktioniert und das Context-Exchange-Plugin als Schnittstelle zur Service-Schicht des GPAPs bereits einsatzfähig ist. Die anschließende Evaluierung des Konzepts der in Abschnitt 3.5.2 vorgestellten heterogenen Adressierung zeigte, dass der GPAP dieses Konzept beispielhaft für die Vermittlung von ICMP-Paketen zwischen IP-basierten Ethernet-Zellen und nicht-IP-basierten Bluetooth-Zellen umsetzt und dabei nur einen geringen Einfluss auf die Latenz und damit auch auf die erzielbare Datenrate hat.

Anschließende Evaluierungen des in Abschnitt 3.5.1 beschriebenen Konzepts zur Service-basierten Kommunikation in heterogenen Netzen zeigten am Beispiel des Service Proxyings zwischen Web Services und Bluetooth-SDP-Diensten eine transparente Kommunikation für die beteiligten Kommunikationsteilnehmer, für die es keinen Unterschied macht, ob sie mit einem Teilnehmer Daten austauschen, der die gleiche oder eine andere Service-Technologie einsetzt, da Dienste jedem Teilnehmer in seiner eigenen Service-Technologie zur Verfügung gestellt werden. Dabei wurde zum einen die generelle Machbarkeit gezeigt. Zum anderen wurden Engpässe bei der Kommunikation der verwendeten Service-Technologien aufgedeckt. Dazu gehört vor allem der in Abbildung 4.11 dargestellte Kommunikations-Overhead bei der Nutzung von Web Services und dem SOAP-Protokoll, das die Service-basierte Kommunikation selbst bei großen Nutzdaten noch mit über 40% Protokoll-Overhead belastet. Der Einfluss des Proxys hat dagegen nur eine geringe Auswirkung auf die während der Service-basierten Kommunikation erzielbare Datenrate. Weiterhin konnte gezeigt werden, dass Service Proxying aber auch in ganz anderen Szenarien wie z.B. der Service-basierten Individualkommunikation zwischen virtuellen und realen Teilnehmern einer Lehrveranstaltung oder einer kontextorientierten Jukebox eingesetzt werden kann, die im Sinne der Adaption intelligenter Umgebungen an die Bedürfnisse der Nutzer, je nach Musikgeschmack der sie umgebenden Personen einen Webradio-Sender auswählt und abspielt, der den meisten gefällt.

Kapitel 5

Zusammenfassung und Ausblick

Die seit 40 Jahren andauernde Entwicklung des Internets hat bis hin zu seiner heutigen Form zu tiefgreifenden Veränderungen für das gesellschaftliche und wirtschaftliche Leben geführt. Mit der Miniaturisierung eingebetteter Systeme und der Entwicklung energiesparsamerer Funktechnologien im Bereich der WPANs entwickelt es sich weiter und es entstehen immer kleinere und leistungsfähigere Geräte, die uns zukünftig verstärkt in unserem alltäglichen Umfeld umgeben und das Potential zur intelligenten, situationsangepassten Assistenz ihrer Nutzer haben. Zur Datenübertragung kommen mit Ethernet, WLAN, Bluetooth und ZigBee schon heute unterschiedliche drahtgebundene und drahtlose Netzwerktechnologien zum Einsatz, die sich in ihrer Reichweite, den zur Verfügung stehenden Datenraten, den Kommunikationsprotokollen und dem Energieverbrauch deutlich unterscheiden.

Die Nutzung der verschiedenen Netzwerktechnologien führte dazu, dass sich das Internet über LANs, WLANs und Mobilfunknetze bis hin zu unseren mobilen Geräten ausweitete und sich zu einem IP-basierten Kernnetz entwickelte. Innerhalb unseres persönlichen Umfelds umgeben uns mit Bluetooth und ZigBee jedoch vorrangig Technologien, die durch eine spontane und direkte Vernetzung kleinerer Endgeräte gekennzeichnet sind und auf Grund ihrer Struktur keine IP-basierte Adressierung benötigen. Gegenüber IP-basierten Netzen setzen sie speziell angepasste Service-Technologien zur Kommunikation ein. Dieser Zwiespalt zwischen IP-basierten sowie nicht-IP-basierten Adressierungs- und Kommunikationsformen, der die Interoperabilität der Netzwerkteilnehmer erschwert bzw. verhindert, wird in unterschiedlichen wissenschaftlichen Arbeiten sowohl mit dem Ansatz einer All-over-IP-basierten Kommunikation (BNEP, 6LoWPAN) auf der Ebene der Internetschicht des TCP/IP-Modells verfolgt, als auch auf der Ebene der Anwendungsschicht durch sogenannte Service-Proxys, die Dienste zwischen verschiedenen Service-Technologien vermitteln.

Der All-over-IP-basierte Ansatz führt aber gerade auf kleinen, mobilen Geräten nicht nur durch die benötigte Zuweisung von IP-, Nameserver- und DNS-Adressen zu einem

Overhead bei der Konfiguration des Netzes, sondern auch beim Datenaustausch zwischen benachbarten WPAN-Geräten, die normalerweise ohne die Nutzung eines IP-Stacks miteinander kommunizieren können. Auch beschränken sich die bisherigen Ansätze des Service Proxyings auf höchstens zwei Service-Technologien und versuchen die Nutzung IP-basierter Service-Technologien in nicht-IP-basierte WPANs durch spezielle Profile auszudehnen, ohne auf die dort vorhandenen Service-Technologien zurückzugreifen, die den Anforderungen einer spontanen, mobilen und ernergiesparsamen Dienstnutzung besser gerecht werden.

5.1 Erreichte Ergebnisse

Diese Arbeit widmete sich dem Problem der Interoperabilität IP-basierter und nicht-IP-basierter Netzwerke sowohl auf der Internetschicht als auch auf der Anwendungsschicht des TCP/IP-Modells. Die in dieser Arbeit vorgestellten Methoden gehen über den Stand der Technik hinaus und ermöglichen eine für die Endgeräte transparente Adressierung und Kommunikation innerhalb eines Ensembles sowie Ensembleübergreifend. Weiterhin erlauben sie eine transparente Integration neuer Netzwerk- und Service-Technologien. Die Kernziele dieser Arbeit wurden in Abschnitt 1.3.2 formuliert und im Rahmen des MuSAMA-Projekts wie folgt umgesetzt:

Konzept zur allgegenwärtigen heterogenen Kommunikation
Aufbauend auf den in aktuellen heterogenen Netzen vorhandenen horizontalen und vertikalen Netzwerkstrukturen wurde das allgemeine Konzept eines zentralisierten General Purpose Access Points vorgestellt. Als Vermittler verbindet er die vorhandenen homogenen Zellen zu einem heterogenen Ensemble und ermöglicht eine allgemeingültige Adressierung und Kommunikation zwischen beliebigen Teilnehmern der beteiligten Zellen. Dabei setzt er mit dem Konzept des Service Proxying und der heterogenen Adressierung sowohl Mechanismen zur Service-basierten als auch zur adressbasierten Kommunikation um. Der Einsatz des GPAPs macht die Nutzung unterschiedlicher Adressierungsarten und SOA-Technologien innerhalb eines Ensembles für die End-Knoten weitestgehend transparent. So können die End-Knoten die für ihre Technologie optimale Adressierungs- (z. B. per MAC-Adressen in Bluetooth- und ZigBee-Netzen) und Kommunikationsmethoden (z. B. Dienstnutzung per Bluetooth SDP) beibehalten und so ressourcenschonend Daten austauschen. Für die End-Knoten ist es so z. B. bei der Suche und Nutzung eines Dienstes transparent, in welcher Technologie er ursprünglich angeboten wurde, da technologiefremde Dienste durch den GPAP in der eigenen Service-Technologie verfügbar gemacht werden und diese SOA-Technologie somit nicht verlassen wird. Die Kopplung mehrerer GPAPs zu

einer Community erlaubt weiterhin die Kommunikation zwischen entfernten Ensembles (z. B. mehreren Smart Labs) und die konzeptionelle Unterstützung der Mobilität von Geräten und deren Diensten innerhalb einer Community.

Referenzarchitektur des GPAPs
Die entwickelte Referenzarchitektur des GPAPs bildet die Basis für ein interoperables und effizientes Kommunikationssystem, das die Heterogenität aktueller pervasiver Umgebungen sowohl auf der Internet- als auch auf der Anwendungsschicht des TCP/IP-Modells systematisch überwindet und eine transparente Einbindung neuer Geräte ermöglicht. Die Referenzarchitektur des GPAPs besteht aus einer Service-Schicht und einer Netzwerkschicht. Die Service-Schicht fokussiert die Gerätekooperation auf der Anwendungsebene und ermöglicht eine Service-basierte Kommunikation nach dem Konzept des Service Proxying zwischen verschiedenen SOAs heterogener Netze. Mit der Netzwerkschicht wurde das Konzept der heterogenen Adressierung am Beispiel der Netzwerktechnologien Bluetooth und Ethernet umgesetzt.

Im Rahmen des GRK MuSAMA wurde mit dieser Arbeit die bereits beschriebene HCBR-Architektur für die Aufgaben der Netzwerkschicht entwickelt. Sie ermöglicht die Integration von drahtgebundenen und drahtlosen Netzwerktechnologien jeglicher Art und übernimmt die Gateway-Funktionalität zwischen den unterschiedlichen Technologien. Durch Realisierung eines Plugin-Konzepts ist die HCBR-Architektur modular mit Device-Plugins und HCBR-Plugins erweiterbar und kann so beliebig um die Unterstützung neuer Netzwerktechnologien sowie Funktionalitäten erweitert werden. Im Rahmen dieser Arbeit wurden die folgenden Funktionsbereiche realisiert:

- Topologie-Bestimmung des GPAP-Verbundes und Kopplung der GPAPs zu einem lokalen Backbone innerhalb einer Community
- Gateway-Funktionalität am Beispiel von Bluetooth- und Ethernet-Zellen
- Schnittstelle für die Zusammenarbeit der Netzwerk- und der Service-Schicht innerhalb eines GPAPs

Die Evaluation der Referenzarchitektur zeigte am Beispiel des Network/Service Visualizers, dass die Zusammenarbeit des HCBR-Cores mit den Device- und HCBR-Plugins problemlos funktioniert und das Context-Exchange-Plugin als Schnittstelle zur Service-Schicht des GPAPs bereits einsatzfähig ist. Weiterhin konnte nachgewiesen werden, dass der GPAP das Konzept der heterogenen Adressierung beispielhaft für die Vermittlung von ICMP-Paketen zwischen IP-basierten Ethernet-Zellen und nicht-IP-basierten Bluetooth-Zellen umsetzt und dabei nur einen geringen Einfluss auf die Latenz und die erzielbare Datenrate hat. Auch bei den Szenarien des Service Proxying zwischen IP-basierten Web Services und nicht-IP-basierten Bluetooth-SDP-Diensten wurde eine für die Kommunikationspartner transparente Kommunikation

durch den GPAP ermöglicht. Weiterhin konnte auch gezeigt werden, dass es auf eine transparente und effiziente Weise möglich ist, Funktionen zur Interoperabilität von Diensten vom GPAP auf z. B. ein Surrogate oder einen beliebigen Server im Internet auszulagern und als Web Service einzubinden. Auch konnte das Konzept des Service Proxyings in ganz anderen Szenarien wie z. B. der Service-basierten Individualkommunikation zwischen virtuellen und realen Teilnehmern einer Lehrveranstaltung oder einer kontextorientierten Jukebox evaluiert werden.

Erweiterbarkeit des Architekturkonzepts
Um eine nahtlose Integration neuer drahtgebundener und drahtloser Netzwerktechnologien zu realisieren, wurden die Device- und HCBR-Plugins der HCBR-Architektur als dynamisch ladbare Shared Object-Bibliotheken implementiert, die sogar zur Laufzeit eine modulare und dynamische Erweiterung des zentralen Bereichs der Netzwerkschicht des GPAPs um die Unterstützung neuer Netzwerktechnologien und Funktionen erlauben. Durch die in der Evaluation verwendete Referenzhardware des GPAPs, die mehrere leistungsfähige USB-Schnittstellen enthält, kann auch eine einfache Erweiterbarkeit der GPAP-Hardware um LAN-, WLAN-, Bluetooth-, ZigBee- und zukünftige Netzwerkschnittstellen erreicht werden. Auf diese Weise werden sowohl der Software als auch der Hardware des GPAPs einfache Erweiterungsmöglichkeiten gegeben, mit der die entwickelte Referenzarchitektur auch den zukünftigen Entwicklungen im Bereich der Netzwerktechnologien Rechnung tragen kann.

5.2 Weiterführende Fragestellungen

Im Rahmen der entwickelten Referenzarchitektur sind Fragen entstanden, die eine Basis für weitere wissenschaftliche Untersuchungen bilden können:

- Für die Integration der zunehmend in smarten Umgebungen eingesetzten ZigBee-Technologie werden sowohl ein ZigBee-Device-Plugin, ein ZigBee-Discoverer-Plugin als auch ein ZigBee-Ethernet-Gateway-Plugin benötigt, die in Anlehnung an die äquivalenten Plugins der bereits unterstützten Bluetooth-Technologie implementiert werden können. So können die in der Nähe der GPAPs befindlichen ZigBee-Geräte mit in die Link-, Neighbor- und DNS-Tabellen der GPAPs eingetragen und in die Topologie des heterogenen Ensembles aufgenommen werden. Weiterhin kann damit die Integration in die vorgestellten Adressierungs- und Kommunikationsmechanismen heterogener Ensembles realisiert werden.
- Die Dauer der Dienstsuche der beteiligten Service-Technologien verhindert derzeit eine unterbrechungsfreie Dienstnutzung. Um eine echtzeitfähige Kommuni-

kation innerhalb einer Community zu erreichen, die zum Großteil aus mobilen Geräten besteht, werden weiterhin Mechanismen zum heterogenen Handover benötigt. Mit dieser Arbeit wurde in Abschnitt 3.6 ein Mobilitätskonzept vorgeschlagen, das die Konzepte der heterogenen Adressierung und des Service Proxyings kombiniert um die Mobilität der End-Knoten und deren Dienste gewährleisten zu können. Zur Realisierung dieses Konzepts sind weiterführende HCBR-Plugins denkbar, die z. B. die Topologie und die Auslastung der Netzwerkverbindungen des heterogenen Netzes sowie Kontextinformationen aus der Service-Schicht nutzen, um ein Handover/Roaming für die mobilen Geräte innerhalb der Community zu realisieren.

- Eine weitere Fragestellung betrifft das Routing in heterogenen Netzwerken. Dabei können die Topologie der Community, die Auslastung der beteiligten Netzwerkschnittstellen und aktuelle Kontextinformationen aus der Service-Schicht für eine geeignete Routing-Metrik für heterogene Netzwerke verwendet werden, um die Netzwerkkapazität in der Community zu erhöhen und möglichst gut zu nutzen. Dazu können messbasierte Verfahren [117] und statusbasierte Verfahren [118] für ein Multi-Path-Routing oder die Nutzung mehrerer Übertragungskanäle innerhalb eines größeren WLANs [119][120][121] mit unterschiedlichen, möglichst nicht überlappenden Funkkanälen betrachtet und eingesetzt werden. Die Informationen aus der Service-Schicht können in der Routing-Metrik aber auch für die Optimierung der Netzwerkpfade zwischen bestimmten Diensten mit Echtzeitanforderungen, wie z. B. einem heterogenen Video- oder Audio-Chat dienen. Zusätzlich kann die beschriebene Betrachtung der logischen Konnektivität eines GPAP-Verbundes zur Einschätzung der Konnektivität und als Orientierungshilfe sowie Ansatzpunkt für eine mögliche Erhöhung der Kapazität des heterogenen Netzwerks genutzt werden. Die gewählte Realisierung der GPAP-Referenzarchitektur in der Programmiersprache C auf Basis eines Linux-Betriebssystems bietet hierzu sogar die Möglichkeit, direkt in das Routing-Verhalten des Betriebssystems einzugreifen und dieses aus einem HCBR-Plugin heraus zu steuern.

Definitionen

Abbildungsverzeichnis

Literaturverzeichnis

[1] Nicholas Carr: *The Big Switch: Rewiring the World, from Edison to Google*. W. W. Norton & Company Ltd., 2008. – ISBN 978–0–393–06228–1

[2] Mark Weiser: The computer for the 21st century. In: *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM, 1999, S. 3–11

[3] Friedemann Mattern: Allgegenwärtige Datenverarbeitung – Trends, Visionen, Auswirkungen. In: *Digitale Visionen*, Springer Berlin Heidelberg, April 2008, S. 3–29

[4] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts and S. Wolff: *A Brief History of the Internet*. `http://www.isoc.org/internet/history/brief.shtml`, 2010

[5] University of Southern California — Information Sciences Institute: *The RFC Editor*. `http://www.rfc-editor.org`,

[6] J. Postel: *Internet Protocol Specification*. `http://tools.ietf.org/html/rfc791`, September 1981

[7] R. Cailliau: *A Little History of the World Wide Web*. `http://www.w3.org/History.html`, 1995

[8] Tim Berners-Lee: *The WorldWideWeb browser*. `http://www.w3.org/People/Berners-Lee/WorldWideWeb`, 1993

[9] European Telecommunications Standards Institute: *Digital cellular telecommunications system (Phase 2+) (GSM); General Packet Radio Service (GPRS); Requirements specification of GPRS (TR 101 186)*. 1999

[10] The 3rd Generation Partnership Project (3GPP): *UMTS-Specifications*. `http://www.3gpp.org/specification-numbering`,

[11] The Working Group for WLAN Standards: *IEEE 802.11TM Wireless Local Area Networks*. `http://grouper.ieee.org/groups/802/11`,

[12] Jochen Schiller: *Mobilkommunikation*. Pearson Studium, 2003. – ISBN 978–3827370600

[13] European Interactive Advertising Association: *EIAA Mediascope Europe 2007*. http://www.eiaa.net/Ftp/casestudiesppt/EIAA_Mediascope_Europe_2007_Pan_European_Executive_Summary.pdf, September 2007

[14] Internet World Stats: *World Internet Usage Statistics - News and World Population Stats*. http://www.internetworldstats.com/stats.htm,

[15] Alan Mauldin: *SEACOM Lights Up Eastern Africa*. www.telegeography.com/mail/seacom_press_2009.html, Juli 2009

[16] Statistisches Bundesamt Deutschland: *Rapider Anstieg der mobilen Internetnutzung durch Unternehmen*. http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/Internet/DE/Presse/pm/2008/04/PD08_163__52911.psml, April 2008

[17] Fittkau & Maass Consulting GmbH: *WWW-Benutzer-Analyse: Ergebnisse der 25. W3B-Umfrage*. http://www.w3b.org/ergebnisse/w3b25/, November 2007

[18] M. Beigl, U. Brinkschulte, F. Feldbusch, D. Fey, S. Fischer, C. Hochberger, R. Hoffmann, W. Karl, J. Krebs, J. Kleinöder, K. Lagally, F. Langhammer, P. Lukowicz, P. Marwedel, E. Maehle, C. Müller-Schloer, B. Schallenberger, H. Schmeck, D. Tavangarian, W. Trumler, T. Ungerer and K. Waldschmidt: *Grand Challenges der Technischen Informatik (Report März 2008)*. http://www.informatik.uni-augsburg.de/de/lehrstuehle/sik/downloads/GC-Report-Maerz08.pdf, März 2008

[19] R. M. Metcalfe and D. R. Boggs: Ethernet: distributed packet switching for local computer networks. In: *Communications of the ACM*, ACM, July 1976, S. 395 – 404

[20] IEEE 802.3 Ethernet Working Group: *IEEE 802.3 - CSMA/CD (ETHERNET)*. http://www.ieee802.org/3/,

[21] ISO/IEC: *Open Systems Interconnection - Basic Reference Model: The Basic Model*. http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip, November 1994

[22] R. Braden: *RFC 1122 - Requirements for Internet Hosts – Communication Layers*. http://www.ietf.org/rfc/rfc1122.txt, Oktober 1989

[23] Joon S. Park and Derrick Dicoi: WLAN Security: Current and Future. In: *Internet Computing Magazine* 7 (2003), September, Nr. 5, S. 60–65

[24] D. Gesbert and J. Akhtar: Breaking the barriers of Shannon's capacity: An overview of MIMO wireless systems. In: *Telenor's Journal: Telektronikk*, Telenor, Januar 2002

[25] T. Clausen and P. Jacquet: *Optimized Link State Routing*. `http://tools.ietf.org/html/rfc3626`, Oktober 2003

[26] C. Perkins, E. Belding-Royer and S. Das: *Ad hoc On-Demand Distance Vector (AODV) Routing*. `http://www.ietf.org/rfc/rfc3561.txt`, Juli 2003

[27] freifunk.net: *Freifunk - freie Netzwerke, freies WLAN, freie Funknetze im deutschsprachigen Raum*. `http://start.freifunk.net/`,

[28] Generation Partnership Project (3GPP), The 3rd: *Homepage*. `http://www.3gpp.org`,

[29] European Information Technology Observatory (EITO): *More than four billion mobile phone users worldwide*. `http://www.eito.com/pressinformation_20090807.htm`, August 2009

[30] Bluetooth Special Interest Group (SIG): *The Official Bluetooth®Technology Info Site*. `http://www.bluetooth.com`,

[31] Bluetooth Special Interest Group: *Bluetooth Specification Version 2.0 + EDR*. `http://german.bluetooth.com/NR/rdonlyres/1F6469BA-6AE7-42B6-B5A1-65148B9DB238/840/Core_v210_EDR.zip`, November 2004

[32] Bluetooth Special Interest Group: *Bluetooth Specification Version 3.0 + HS*. `http://german.bluetooth.com/NR/rdonlyres/298BE70B-4353-4492-9A91-160549463612/10885/Core_V30__HS.zip`, April 2009

[33] ZigBee Alliance: *The Official ZigBee Technology Info Site*. `http://www.zigbee.org`,

[34] ZigBee Alliance: *ZigBee Specification*. `http://zigbee.org/ZigBeeSpecificationDownloadRequest/tabid/311/Default.aspx`, 2008

[35] H. Labiod, H. Afifi and C. D. Santis: *Wi-Fi, Bluetooth, ZigBee and WiMax*. Springer, 2007. – ISBN 978–1–4020–5396–2

[36] G. Thonet, P. Allard-Jacquin and P. Colle: *ZigBee – WiFi Coexistence: White Paper and Test Report*. `http://www.zigbee.org/en/press_kits/2009_07_01/documents/white_papers/wp_zigbeewifi_final.pdf`, April 2008

[37] P. M. Bull, P. R. Benyon and P. R. Limb: Residential Gateways. In: *BT Technology Journal*, Springer Netherlands, April 2002, S. 73–81

[38] KNX Association: *KNX System Specifications — Architecture.* http://www.knx.org/fileadmin/downloads/03-KNXStandard/KNXStandardPublicDocuments/03_01_01Architecturev3.0.zip, Juni 2009

[39] D. Jensen, A. Schmidt and M. Zeidler: Combining the KNX and an IEEE 802.15.4 based wireless system to build a Context Aware System. In: *KNX Scientific Conference*, 2005

[40] Bundesministerium für Bildung und Forschung: *Ambient Assisted Living - Rahmenprogramm Mikrosysteme.* http://www.aal-deutschland.de, 2009

[41] Fraunhofer-Gesellschaft: *Förderprojekt SmarterWohnen®NRW.* http://www.smarterwohnen.net, 2009

[42] Technische Universität Kaiserslautern: *Assisted Living - Selbstbestimmtes Wohnen im Alter mit moderner Technik.* http://www.eit.uni-kl.de/litz/assisted_living, Juni 2008

[43] DAI-Labor der Technischen Universtität Berlin: *Service Centric Home (SerCHo) - Showroom.* http://130.149.154.94/index.php?id=49, Juni 2007

[44] SmartHome Paderborn: *SmartHome - So lebt man heute.* http://www.smarthomepaderborn.de, 2009

[45] Haus der Gegenwart GmbH: *Architektur, Design und Wohnen im Haus der Gegenwart.* http://www.haus-der-gegenwart.de, 2010

[46] Fraunhofer-inHaus-Zentrum: *inHaus - Innovationszentrum der Fraunhofer-Gesellschaft.* http://www.inhaus-zentrum.de, 2010

[47] Thomas Kirste - MuSAMA Rostock: *Multimodal Smart Appliance Ensembles for Mobile Applications.* http://www.informatik.uni-rostock.de/musama.html, 2010

[48] Bundesamt für Sicherheit in der Informationstechnik: *Drahtlose Kommunikationssysteme und ihre Sicherheitsaspekte.* https://www.bsi.bund.de/cae/servlet/contentblob/487312/publicationFile/42808/drahtkom_pdf.pdf, September 2009

[49] RFC-Editor Homepage: *Official Internet Protocol Standards.* http://www.rfc-editor.org/rfcxx00.html, Februar 2010

[50] Registration Authority: *Guidelines for 64-Bit Global Identifier (EUI-64).* http://standards.ieee.org/regauth/oui/tutorials/EUI64.html,

[51] J. Postel: *Transmission Control Protocol.* http://www.ietf.org/rfc/rfc793.txt, September 1981

[52] J. Postel: *User Datagram Protocol.* http://www.ietf.org/rfc/rfc768.txt, August 1980

[53] K. Egevang and P. Francis: *The IP Network Address Translator (NAT).* http://tools.ietf.org/html/rfc1631, Mai 1994

[54] P. Srisuresh and K. Egevang: *Traditional IP Network Address Translator (Traditional NAT).* http://tools.ietf.org/html/rfc3022, Januar 2001

[55] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot and E. Lear: *Address Allocation for Private Internets.* http://www.faqs.org/ftp/rfc/rfc1918.txt, Februar 1996

[56] R. Finlayson, T. Mann, J. Mogul and M. Theimer: *A Reverse Address Resolution Protocol.* http://tools.ietf.org/html/rfc903, Juni 1984

[57] B. Croft and J. Gilmore: *Bootstrap Protocol (BOOTP).* http://tools.ietf.org/html/rfc951, September 1985

[58] R. Droms: *Dynamic Host Configuration Protocol.* http://tools.ietf.org/html/rfc2131, März 1997

[59] R. Braden: *Requirements for Internet Hosts - Application and Support.* http://tools.ietf.org/html/rfc1123, Oktober 1989

[60] P. Mockapetris: *Domain Names - Implementation and Specification.* http://tools.ietf.org/html/rfc1035, November 1987

[61] D. C. Plummer: *An Ethernet Address Resolution Protocol.* http://tools.ietf.org/html/rfc826, November 1982

[62] World Wide Web Consortium (W3C): *Web Services Addressing 1.0 - SOAP Binding.* http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/, May 2006

[63] World Wide Web Consortium (W3C): *SOAP Version 1.2 Part 1: Messaging Framework.* http://www.w3.org/TR/2007/REC-soap12-part1-20070427, April 2007

[64] World Wide Web Consortium (W3C): *Web Services Description Language (WSDL) Version 2.0.* http://www.w3.org/TR/wsdl20/, Juni 2007

[65] L. Clement, A. Hately, C. v. Riegen and T. Rogers: *The UDDI Version 2 OASIS Standard.* `http://www.oasis-open.org/specs/index.php#uddiv3.0.2`, Februar 2005

[66] K. Ballinger, P. Brittenham, A. Malhotra, W. A. Nagy and S. Pharies: *Web Services Inspection Language (WS-Inspection) 1.0.* `http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-wsilspec/ws-wsilspec.pdf`, November 2001

[67] P. Leach, M. Mealling and R. Salz: *A Universally Unique IDentifier (UUID) URN Namespace.* `http://tools.ietf.org/rfc/rfc4122.txt`, Juli 2005

[68] LAN/MAN Standards Committee: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).* `http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf`, September 2006

[69] Bluetooth Special Interest Group: *Bluetooth Network Encapsulation Protocol (BNEP) Specification.* `http://grouper.ieee.org/groups/802/15/Bluetooth/BNEP.pdf`, Juni 2001

[70] Bluetooth Special Interest Group: *Personal Area Networking Profile (PAN).* `http://bluetooth.com/NR/rdonlyres/36395112-9A8F-44B9-8332-DDDC4FFF51C6/984/PAN_SPEC_V11.pdf`, Februar 2003

[71] C. Cordeiro, S. Abhyankar, R. Toshiwal and D. Agrawal: BlueStar: Enabling Efficient Integration between Bluetooth WPANs and IEEE 802.11 WLANs. In: *Mobile Networks and Applications (MONET)*, Kluwer Academic Publishers, August 2004, S. 409–422

[72] M.T. Zia, M.U. Farooq and S.A. Khan: Seamless Communication over Heterogeneous Interfaces in Mobile Ad Hoc Networks. In: *Proceedings of the Sixth international conference on Wireless and Optical Communications Networks (WOCN*, IEEE Press, 2009, S. 329–333

[73] S. Pinkumphi and A. Phonphoem: Real-Time Audio Multicasting on Bluetooth Network. In: *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, IEEE Computer Society, Mai 2009, S. 992 – 995

[74] M.J. Moron, R. Luque, E. Casilari and A. Diaz-Estrella: Analysis of Bluetooth Transmission Delay in Personal Area Networks. In: *3rd International Symposium on Wireless Pervasive Computing (ISWPC)*, IEEE Computer Society, Mai 2008, S. 620 – 622

[75] Internet Engineering Task Force: *IPv6 over Low power WPAN (6lowpan)*. `http://www.ietf.org/dyn/wg/charter/6lowpan-charter.html`,

[76] N. Kushalnagar, G. Montenegro and C. Schumacher: *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*. `http://www.ietf.org/rfc/rfc4919.txt`, August 2007

[77] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler: *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. `http://www.ietf.org/rfc/rfc4944.txt`, September 2007

[78] M. Durvy, J. Abeillé, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne and A. Dunkels: Making Sensor Networks IPv6 Ready. In: *Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys)*, ACM, 2008, S. 421–422

[79] Reen-Cheng Wang, Ruay-Shiung Chang and Han-Chieh Chao: Internetworking Between Zigbee/802.15.4 and IPv6/802.3 Network. In: *SIGCOMM 2007 Workshop: IPv6 and the Future of the Internet*, ACM, 2007

[80] Karl Mayer and Wolfgang Fritsche: IP-enabled Wireless Sensor Networks and their integration into the Internet. In: *Proceedings of the first international conference on Integrated internet ad hoc and sensor networks (InterSense)*, ACM, 2006

[81] Flutra Osmani and Adriaan Slabbert: A scalable distributed security infrastructure for industrial control and sensor networks. In: *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly (IWCMC)*, ACM, 2009, S. 84–89

[82] David Culler: *Secure, low-power, IP-based connectivity with IEEE 802.15.4*. `http://www.archrock.com/downloads/resources/ArchRock.Sum07.pdf`, 2007

[83] On Shun Chau, Pan Hui and V. O. K. Li: An Architecture Enabling Bluetooth™/JINI™ Interoperability. In: *15th IEEE International Symposium*

on Personal, Indoor and Mobile Radio Communications (PIMRC), IEEE Computer Society, September 2004, S. 3013– 3018

[84] A. Delphinanto, A.M.J. Koonen, M.E. Peeters and F.T.H. den Hartog: Proxying UPnP service discovery and access to a non-IP Bluetooth network on a mobile phone. In: *14th IEEE Symposium on Communications and Vehicular Technology*, IEEE Computer Society, November 2007, S. 1 – 5

[85] V. Auletta, C. Blundo, E. De Cristofaro and G. Raimato: Performance Evaluation of Web Services Invocation over Bluetooth. In: *Proceedings of the ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks*, ACM, 2006, S. 1–8

[86] D. S. Mackie: *Extending the Reach of Personal Area Networks by Transporting Bluetooth Communications Over IP Networks*. `http://eprints.ru.ac.za/861/01/mackie-msc-tr07-23.pdf`, Dezember 2006

[87] Young-Guk Ha: Dynamic Integration of ZigBee Home Networks into Home Gateways Using OSGi Service Registry. In: *IEEE Transactions on Consumer Electronics*, IEEE Computer Society, Mai 2009, S. 470 – 476

[88] Seong Hoon Kim, Jeong Seok Kang, Hong Seong Park, Daeyoung Kim and Young-Joo Kim: UPnP-ZigBee Internetworking Architecture Mirroring a Multi-hop ZigBee Network Topology. In: *IEEE Transactions on Consumer Electronics*, IEEE Computer Society, August 2009, S. 1286 – 1294

[89] Jeong-Hee Kim, Do-Hyeon Kim, Ho-Young Kwak and Yung-Cheol Byun: Address Internetworking between WSNs and Internet supporting Web Services. In: *International Conference on Multimedia and Ubiquitous Engineering (MUE)*, IEEE Computer Society, April 2007, S. 232–240

[90] N. B. Priyantha, A. Kansal, M. Goraczko and F. Zhao: Tiny Web Services: Design and Implementation of Interoperable and Evolvable Sensor Networks. In: *Proceedings of the 6th ACM conference on Embedded Network Sensor Systems (SenSys)*, ACM, 2008, S. 253–266

[91] G. Moritz, E. Zeeb, F. Golatowski, D. Timmermann and R. Stoll: Web Services to Improve Interoperability of Home Healthcare Devices. In: *3rd International Conference on Pervasive Computing Technologies for Healthcare*, 2009

[92] Organization for the Advancement of Structured Information Standards (OASIS): *Devices Profile for Web Services Version 1.1*. `http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.html`, Juli 2009

[93] Heiko Kopp: *Design und Management selbstorganisierender drahtloser Backbone-Netzwerke*, Universität Rostock, Dissertation, 2009

[94] Dirk Hachenberger: *Mathematik für Informatiker*. Pearson Studium, 2005. – ISBN 978–3–8273–7109–6

[95] Andreas Brandstädt: *Graphen und Algorithmen*. Teubner Verlag, 1994. – ISBN 978–3–519–02131–5

[96] M. E. J. Newman: The Mathematics of Networks. In: *The New Palgrave Encyclopedia of Economics, 2nd edition*, Palgrave Macmillan, Mai 2008

[97] Roger W. Hockney: *The Science of Computer Benchmarking (Software, Environments, Tools)*. Society for Industrial & Applied Mathematics (SIAM), U.S., 1996. – ISBN 978–0898713633

[98] E. Dressler, R. Zender, U. Lucke and D. Tavangarian: A new Architecture for Heterogeneous Context Based Routing. In: *Advances in Computer Science and Engineering - 13th International CSI Computer Conference (CSICC)*, Springer Berlin Heidelberg, März 2008, S. 526–534

[99] E. Dressler, R. Zender, U. Lucke and D. Tavangarian: A Multi-Layer Approach for Cross-Technology Communication in a Pervasive Community. In: *The Sixth Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2009

[100] OpenWrt: *Linux distribution for embedded devices*. `http://www.openwrt.org`,

[101] Linutop SARL: *Linutop - Mini Linux PC*. `http://www.linutop.com/linutop2/index.de.html`,

[102] R. Zender, E. Dressler, U. Lucke and D. Tavangarian: Meta-Service Organization for Pervasive Universities. In: *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2008, S. 400–405

[103] Jui-Hao Chiang and Tzi-cker Chiueh: Implementation and Evaluation of a Mobile Tetherless VoIP/PSTN Gateway. In: *6th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, IEEE Computer Society, Juli 2009

[104] LAN/MAN Standards Committee: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*. `http://standards.ieee.org/getieee802/download/802.15.1-2005.pdf`, Juni 2005

[105] Bluetooth Special Interest Group: *Bluetooth Assigned Numbers*. http://bluetooth.com/NR/rdonlyres/CB9E945E-F4D5-4CB1-ACB3-0261892DD6C5/10446/AssignedNumbersOverview.pdf, Februar 2009

[106] N. Nasser, A. Hasswa and H. Hassanein: Handoffs in fourth generation heterogeneous networks. In: *IEEE Communications Magazine* 44 (2006), Oktober, Nr. 10

[107] H. Liu, C. Maciocco and V. Kesavan: Using Predictive Triggers to Improve Handover Performance in Mixed Networks. In: *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, Springer Berlin / Heidelberg, Mai 2008, S. 877–888

[108] M.L. George, L.J. Kallidukil and Jong-Moon Chung: Bluetooth Handover Control for Roaming System Applications. In: *Proceedings of the 45th Midwest Symposium on Circuits and Systems (MWSCAS)*, IEEE Computer Society, August 2002, S. 404–407

[109] Ming-Chiao Chen, Jiann-Liang Chen and Pei-Chun Yao: Efficient handoff algorithm for Bluetooth networks. In: *IEEE International Conference on Systems, Man and Cybernetics*, IEEE Computer Society, Oktober 2005, S. 3884 – 3889

[110] R. Zender and D. Tavangarian: Service-oriented University: Infrastructure for the University of Tomorrow. In: *Intelligent Interactive Assistance and Mobile Multimedia Computing*, Springer Berlin Heidelberg, November 2009, S. 73–84

[111] The tcpdump/libpcap project: *User level packet capturing*. http://www.tcpdump.org,

[112] The libnet project: *User level packet construction*. http://libnet.sourceforge.net,

[113] BlueZ: *Official Linux Bluetooth protocol stack*. http://www.bluez.org,

[114] H. Pennington, A. Carlsson and A. Larsson: *D-Bus Specification*. http://dbus.freedesktop.org/doc/dbus-specification.html, August 2007

[115] BlueZ: *D-Bus API description and examples*. http://wiki.bluez.org,

[116] Harald Welte: *The netfilter.org libnetfilter_queue project*. http://www.netfilter.org/projects/libnetfilter_queue/index.html,

[117] K. Liu, J. Li, P. Huang and A. Fukuda: Adaptive Acquisition Multiple Access Protocol in Wireless Multihop Mobile Ad Hoc Networks. In: *IEEE 55th Vehicular Technology Conference (VTC)*, IEEE Computer Society, August 2002, S. 60–64

[118] J. LI, Z. Haas, M. Sheng and Y. Chen: Performance Evaluation of Modified IEEE 802.11 MAC for Multi-Channel Multi-Hop Ad Hoc Networks. In: *17th International Conference on Advanced Information Networking and Applications (AINA)*, IEEE Computer Society, März 2003, S. 312–317

[119] R. Draves, J. Padhye and B. Zill: Routing in multi-radio, multi-hop wireless mesh networks. In: *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom)*, ACM, 2004, S. 114–128

[120] C. Chereddi, P. Kyasanur and N. H. Vaidya: Design and implementation of a multi-channel multi-interface network. In: *Proceedings of the 2nd international workshop on multi-hop ad hoc networks: from theory to reality (REALMAN*, ACM, 2006, S. 23–30

[121] A. A. Pirzada, M. Portman and J. Indulska: Evaluation of multi-radio extensions to AODV for wireless mesh networks. In: *Proceedings of the 4th ACM international workshop on mobility management and wireless access (MOBIWAC)*, ACM, 2006, S. 45–51

[122] J. Postel: *Internet Control Message Protocol*. `http://tools.ietf.org/rfc/rfc0792.txt`, September 1981

[123] Enrico Dressler: *Konzeption und Entwicklung eines echtzeitfähigen Protokoll-Konverters zur adaptiven Datenweiterleitung*, Universität Rostock, Diplomarbeit, Mai 2006

[124] Friedrich Meincke: *Konzept und Implementierung eines Bluetooth-IP-Proxys*, Universität Rostock, Masterarbeit, November 2008

[125] avetana GmbH: *avetana Bluetooth JSR-82 Implementation*. `http://www.avetana-gmbh.de/avetana-gmbh/produkte/jsr82.eng.xml`,

[126] JDOM Project: *Java Document Object Model (JDOM)*. `http://www.jdom.org/index.html`,

[127] E. Dressler and D. Tavangarian: Heterogeneous Communication in Smart Ensembles. In: *Intelligent Interactive Assistance and Mobile Multimedia Computing*, Springer Berlin Heidelberg, November 2009, S. 155–166

[128] P. Lehsten, A. Thiele, R. Zilz, E. Dressler, R. Zender, U. Lucke and D. Tavangarian: Dienste-basierte Kopplung von virtueller und Präsenzlehre. In: *DeLFI 2008: Die 6. e-Learning Fachtagung Informatik*, Bonner Köllen Verlag, September 2008

[129] Linden Reseach, Inc: *Second Life Official Site*. `http://secondlife.com`,

[130] P. Lehsten and T. Reichelt: *Servicebasierte Individualkommunikation zwischen Teilnehmern von virtuellen und Präsenzlehrveranstaltungen*, Universität Rostock, Projektarbeit, Mai 2009

[131] Dennis Lenz: *Kontextorientierte SOA-Interoperabilität für Broadcast-Szenarien*, Universität Rostock, Diplomarbeit, Oktober 2009

[132] BlueCove: *BlueCove JSR-82 Project*. `http://bluecove.org/`,

[133] VLC Media Player: *Open Source Multimedia Framework and Player*. `http://www.videolan.org/vlc/`,

Abkürzungsverzeichnis

3G ..... 3. Mobilfunkgeneration
3GPP ..... 3rd Generation Partnership Project
4G ..... 4. Mobilfunkgeneration
6LoWPAN ..... IPv6 over Low power WPAN
ACL ..... Asynchronous Connectionless Link
AM ..... Adjazenzmatrix
AODV ..... Ad-hoc On-demand Distance Vector
AP ..... Access Point
APL ..... Application Layer
APS ..... Application Support Sublayer
ARP ..... Address Resolution Protocol
ARPANET .... Advanced Research Projects Agency Network
BFS ..... Breadth-First Search
BNEP ..... Bluetooth Network Encapsulation Protocol
BOOTP ..... Bootstrap Protocol
BPSK ..... Binary Phase Shift Keying
BWG ..... Bluetooth Wireless Gateway
CAC ..... Channel Access Code
CID ..... Channel Identifier
CSMA/CA .... Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD .... Carrier Sense Multiple Access with Collision Detection
DFG ..... Deutsche Forschungsgemeinschaft
DFS ..... Depth-First Search
DHCP ..... Dynamic Host Configuration Protocol
DHCPCD ..... DHCP Client Daemon
DNAT ..... Destination Network Address Translation
DNS ..... Domain Name System
DPWS ..... Device Profile for Web Services
DSSS ..... Direct Sequence Spread Spectrum

EDGE ..... Enhanced Data Rates for GSM Evolution
EDR ..... Enhanced Data Rate
EIB ..... European Installation Bus
EIBA ..... European Installation Bus Association
EK ..... End-Knoten
ETSI ..... European Telecommunications Standards Institute
EUI ..... Extended Unique Identifier
FFD ..... Full Function Device
FTP ..... File Transfer Protocol
GCC ..... GNU Compiler Collection
GK ..... Gateway-Knoten
GPAP ..... General Purpose Access Point
GPRS ..... General Packet Radio Service
GSM ..... Global System for Mobile Communications
GUI ..... Graphical User Interface
GUID ..... Globally Unique Identifier
HCBR ..... Heterogeneous Context-based Routing
HCI ..... Host Controller Interface
HSPA ..... High Speed Packet Access
HTTP ..... HyperText Transfer Protocol
HTTPS ..... HyperText Transfer Protocol Secure
ICMP ..... Internet Control Message Protocol
IEEE ..... Institute of Electrical and Electronics Engineers
IETF ..... Internet Engineering Task Force
IMT-2000 ..... International Mobile Telecommunications-2000
IP ..... Internet Protocol
IPC ..... Inter-Process Communication
IPSec ..... Internet Protocol Security
ISDN ..... Integrated Services Digital Network
ISM ..... Industrial, Scientific, and Medical
ISO ..... International Organization for Standardization
JDK ..... Java Development Kit
JNI ..... Java Native Interface
JRE ..... Java Runtime Environment
L2CAP ..... Logical Link Control and Adaptation Protocol

LAN ..... Local Area Network
LAP ..... Lower Address Part
LTE ..... Long Term Evolution
MAC ..... Media Access Control
MIMO ..... Multiple Input Multiple Output
MIT ..... Massachusetts Institute of Technology
MuSAMA ..... Multimodal Smart Appliance Ensembles for Mobile Applications
NAP ..... Network Access Point
NAP ..... Non-significant Address Part
NAPT ..... Network Address Port Translation
NAT ..... Network Address Translation
NDP ..... Neighbor Discovery Protocol
NGMN ..... Next Generation Mobile Network
NGN ..... Next Generation Network
NWK ..... Network Layer
OBEX ..... Object Exchange Protocol
OFDM ..... Orthogonal Frequency Division Multiplex
OLSR ..... Optimized Link State Routing
OQPSK ..... Offset Quadrature Phase Shift Keying
OSGi ..... Open Services Gateway initiative
OSI ..... Open Systems Interconnection
OSI-Modell .... Open Systems Interconnection Reference Model
OUI ..... Organizationally Unique Identifier
P2P ..... Peer-to-Peer
PAN ..... Personal Area Network
PANU ..... Personal Area Network User
PARC ..... Xerox Palo Alto Research Center
PAT ..... Port Address Translation
PC ..... Personal Computer
PDA ..... Personal Digital Assistant
PHY ..... Physical Layer
PSTN ..... Public Switched Telephone Network
QoS ..... Quality of Service
RARP ..... Reverse Address Resolution Protocol
RFC ..... Requests for Comments

RFCOMM ..... Radio Frequency Communication
RFD ..... Reduced Function Device
RPC ..... Remote Procedure Call
RSSI ..... Received Signal Strength Indication
SCO ..... Synchronous Connection Oriented Link
SDAP ..... Service Discovery Application Profile
SDDB ..... Service Discovery Database
SDP ..... Service Discovery Protocol
SIG ..... Bluetooth Special Interest Group
SIP ..... Session Initiation Protocol
SL ..... Second Life
SM ..... Schnittstellenmatrix
SNAT ..... Source Network Address Translation
SNR ..... Signal to Noise Ratio
SOA ..... Service-orientierte Architektur
SOAP ..... Simple Object Access Protocol
SR ..... Service Record
SSID ..... Service Set Identifier
SSM ..... STIL Service Manager
STIA ..... Service Technology Independent Architecture
STIL ..... Service Technology Independent Language
STP ..... Service Technology Plugins
STT ..... Service Technology Translators
SV ..... Schnittstellenvektor
TCP ..... Transmission Control Protocol
TLV ..... Tag Length Value
ToS ..... Type of Service
TTL ..... Time To Live
UAP ..... Upper Address Part
UDDI ..... Universal Description, Discovery and Integration
UDP ..... User Datagram Protocol
UMTS ..... Universal Mobile Telecommunications System
UPnP ..... Universal Plug and Play
UUID ..... Universally Unique Identifier
VoIP ..... Voice over IP

VPN ..... Virtual Private Network
WCDMA ..... Wideband CDMA
WLAN ..... Wireless Local Area Network
WPA2 ..... Wi-Fi Protected Access 2
WPAN ..... Wireless Personal Area Network
WS ..... Web Service
WSDL ..... Web Services Description Language
WSIL ..... Web Services Inspection Language
WTK ..... Wireless Toolkit
WWW ..... World Wide Web
ZDO ..... ZigBee Device Object

Stichwortverzeichnis

Veröffentlichungen und Fachvorträge

Publikationen

1. Enrico Dressler.
   Konzeption und Entwicklung eines echtzeitfähigen Protokoll-Konverters zur adaptiven Datenweiterleitung.
   Diplomarbeit, Universität Rostock, Fachbereich Informatik, 2006.

2. D. Tavangarian, K. Nölting, C. C. Schnekenburger und E. Dressler.
   E-Learning in Mecklenburg-Vorpommern. Zum aktuellen Stand der E-Learning-Aktivitäten.
   Studie im Auftrag des Ministeriums für Bildung, Wissenschaft und Kultur Mecklenburg-Vorpommern, 2007.

3. E. Dressler, R. Zender, U. Lucke und D. Tavangarian.
   A new Architecture to Realise Pervasive Communities.
   7th International Workshop on Innovative Internet Community Systems (I2CS), München, Juni 2007.

4. E. Dressler, Raphael Zender, U. Lucke und D. Tavangarian.
   Eine neuartige Architektur zur Realisierung von Pervasive Communitys.
   2. Workshop Pervasive University im Rahmen der 37. GI Jahrestagung, Bremen, September 2007.

5. R. Zender, E. Dressler, U. Lucke und D. Tavangarian.
   Meta-Service Organization for a Pervasive University.
   PerEL Workshop at 7th IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE Computer Society, Hong Kong, März 2008.

6. E. Dressler, Raphael Zender, U. Lucke und D. Tavangarian.
   A new Architecture for Heterogeneous Context Based Routing.
   13th International CSI Computer Conference, Springer, Kish Island, Iran, März 2008.

7. P. Lehsten, A. Thiele, R. Zilz, E. Dressler, R. Zender, U. Lucke und D. Tavangarian.
   Dienste-basierte Kopplung von virtueller und Präsenzlehre.
   6. e-Learning Fachtagung Informatik der Gesellschaft für Informatik (DeLFI 2008), Lübeck, September 2008.

8. R. Zender, E. Dressler, U. Lucke und D. Tavangarian.
   Pervasive Media and Messaging Services for Immersive Learning Experiences.
   5th IEEE International Workshop on PervasivE Learning, Galveston, März 2009.

9. E. Dressler, R. Zender, U. Lucke und D. Tavangarian.
   A Multi-Layer Approach for Cross-Technology Communication in a Pervasive Community.
   Sixth Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Toronto, Juli 2009.

10. E. Dressler, D. Tavangarian.
    Heterogeneous Communication in Smart Ensembles.
    In Proceedings of the Intelligent Interactive Assistance and Mobile Multimedia Computing (IMC), Rostock-Warnemünde, November 2009.

Fachvorträge

1. E. Dressler.
   Eine neuartige Architektur zur Realisierung von Pervasive Communitys.
   2. Workshop Pervasive University im Rahmen der 37. GI Jahrestagung, Bremen, September 2007.

2. E. Dressler.
   A new Architecture for Heterogeneous Context Based Routing.
   13th International CSI Computer Conference, Springer, Kish Island, Iran, März 2008.

3. E. Dressler.
   A Multi-Layer Approach for Cross-Technology Communication in a Pervasive Community.
   Sixth Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Toronto, Juli 2009.

4. E. Dressler.
   Heterogeneous Communication in Smart Ensembles.
   In Proceedings of the Intelligent Interactive Assistance and Mobile Multimedia Computing (IMC), Rostock-Warnemünde, November 2009.

5. E. Dressler.
   Fortschrittsberichte im Rahmen des Graduiertenseminars des GRK MuSAMA. Am 08.03.2007, 06.12.2007, 29.04.2008, 24.06.2008 und 24.11.2009.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbstständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel angefertigt habe. Die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen sind als solche kenntlich gemacht.

Ich versichere weiterhin, dass ich bisher weder die vorliegende Dissertation noch Teile von ihr als Prüfungsarbeit oder zum Zweck der Promotion eingereicht bzw. verwendet habe.

Rostock, 08. April 2010

Enrico Dressler

Dipl.-Inf. Enrico Dressler

Geburtsdatum	02.11.1979
Geburtsort	Anklam
Familienstand	ledig
Staatsangehörigkeit	deutsch
Sprachen	Deutsch (Muttersprache) Englisch (fließend in Wort und Schrift)

Lebenslauf

09/1986 - 07/1990	Besuch der Grundschule in Friedland
08/1990 - 07/1992	Besuch des Gymnasiums in Friedland
08/1992 - 07/1998	Sportgymnasium Neubrandenburg mit Abschluss allgemeine Hochschulreife, Hauptfächer Mathematik und Physik
11/1998 - 08/1999	Wehrdienst
10/1999 - 05/2006	Studium der Informatik, Universität Rostock
03/2002 - 11/2002	studentische Hilfskraft am Fraunhofer Institut für Graphische Datenverarbeitung, Rostock
08/2003 - 01/2004	Studienarbeit mit dem Thema „Optimierung der Code-Erzeugung für Java JIT-Compiler“
09/2003 - 12/2005	studentische Hilfskraft am Lehrstuhl für Rechnerarchitektur, Universität Rostock
04/2005 - 07/2005	Praktikum beim Micon Start-Up-Labor, Rostock
11/2005 - 05/2006	Diplomarbeit mit dem Thema „Konzeption und Entwicklung eines echtzeitfähigen Protokoll-Konverters zur adaptiven Datenweiterleitung“
05/2006	Abschluss des Studiums als Diplom-Informatiker
07/2006 - 10/2006	Wissenschaftlicher Projektmitarbeiter am Lehrstuhl für Rechnerarchitektur, Universität Rostock
10/2006 - 01/2010	Promotionsstudent im Rahmen des Graduiertenkollegs „Multimodal Smart Appliance Ensembles for Mobile Applications“ (MuSAMA, GRK 1424/1), gefördert von der Deutschen Forschungsgemeinschaft (DFG)

Thesen

1. Der anhaltende Fortschritt der Mikroelektronik führt zu immer kleineren, schnelleren und energieeffizienteren Netzwerkschnittstellen, die den Einsatz in kleinen sowie spontan und drahtlos miteinander kommunizierenden Geräten (z. B. Sensoren, Mobiltelefonen und Notebooks) ermöglichen und sich zunehmend in unseren alltäglichen Gegenständen verstecken.

2. Geräte, die sich in unserer unmittelbaren Umgebung befinden und mit integrierter Kommunikationsfähigkeit sowie Sensoren ausgestattet sind, nehmen schon heute vielfältige Umgebungsinformationen wahr. Sie haben das technische Potential diese Informationen miteinander auszutauschen und mit Hilfe von Aktoren auf spezielle Ereignisse in der Umgebung zu reagieren. Die proaktive und situationsangepasste Unterstützung des Nutzers steht dabei im Fokus.

3. Gerade im Bereich der intelligenten Umgebungen und des *Ambient Assisted Living* (AAL) treffen verschiedene drahtgebundene und drahtlose Netzwerktechnologien mit unterschiedlichen Eigenschaften (z. B. Reichweite, Bandbreite, Energieverbrauch, Adressierungsart und Kommunikationsprotokoll) aufeinander, die eine heterogene Kommunikation zwischen beliebigen Geräten bisher erschwert bzw. verhindert.

4. Neben Netzwerktechnologien wie z. B. Ethernet, WLAN, WiMAX und UMTS, die vorrangig der Kommunikation mit dem Internet dienen und daher auf der Nutzung des Internet Protokolls basieren, ermöglichen *Wireless Personal Area Networks* (WPAN) auf Grund ihrer Struktur eine IP-freie, spontane und direkte Vernetzung von Endgeräten und verlängern die Akkulaufzeit der mobilen Geräte durch den Einsatz geringer Sendeleistungen sowie spezieller, energiesparsamer Adressierungs- und Kommunikationsmechanismen deutlich.

5. Der aktuelle Stand der Forschung zur Interoperabilität von WPAN-Technologien und IP-basierten Netzwerktechnologien führt mit dem Ansatz einer All-over-IP-basierten Kommunikation auf der Internetschicht des TCP/IP-Referenzmodells zu einer Erweiterung der Protokoll-Stacks der WPAN-Technologien um die IP-Funktionalität. Gerade auf kleinen, mobilen Geräten stellt dieser Ansatz durch die nun benötigte Zuweisung von IP-, Nameserver- und DNS-Adressen einen

nicht zu vernachlässigbaren Overhead bei der Konfiguration des Netzwerks sowie einen Protokoll-Overhead während der Kommunikation dar.

6. Weitere bisherige Ansätze zur Lösung der Interoperabilitätsprobleme von WPAN-Technologien und IP-basierten Netzwerktechnologien nutzen das Konzept des Service Proxying auf der Anwendungsschicht des TCP/IP-Referenzmodells, das bisher auf höchstens zwei Service-Technologien beschränkt ist und die Nutzung IP-basierter Service-Technologien in nicht-IP-basierte WPANs durch spezielle Profile ausdehnt, ohne die dort vorhandenen Service-Technologien zu berücksichtigen.

7. Der mit dieser Arbeit entwickelte *General Purpose Access Point* (GPAP) erweitert das Konzept des Service Proxying und setzt mit dem Konzept der heterogenen Adressierung sowohl Mechanismen zur Service-basierten als auch zur adressbasierten Kommunikation um. Dabei überwindet er die Interoperabilitätsprobleme verschiedener IP- und nicht-IP-basierter Netzwerktechnologien und macht die Nutzung unterschiedlicher Adressierungsarten und SOA-Technologien, sowohl innerhalb eines Ensembles als auch Ensemble-übergreifend in einer Community, für die End-Knoten weitestgehend transparent.

8. Der GPAP besteht aus einer Service-Schicht und einer Netzwerkschicht. Die Service-Schicht fokussiert die Gerätekooperation auf der Anwendungsebene und ermöglicht eine Service-basierte Kommunikation zwischen den verschiedenen SOA-Technologien der unterschiedlichen Netzwerktechnologien. Die Netzwerkschicht setzt das Konzept der heterogenen Adressierung um und ermöglicht die Integration von drahtgebundenen und drahtlosen Netzwerktechnologien jeglicher Art. Somit übernimmt sie die Gateway-Funktionalität zwischen den unterschiedlichen Technologien.

9. Die Referenzarchitektur des GPAPs bildet die Basis für ein interoperables und effizientes Kommunikationssystem, das die Heterogenität aktueller pervasiver Umgebungen (z. B. Smart Homes, Smart Ensembles) sowohl auf der Service- als auch auf der Netzwerk-Ebene systematisch überwindet und eine transparente Einbindung neuer Geräte ermöglicht.

10. Der modulare Aufbau der GPAP-Architektur ermöglicht eine nahtlose Integration neuer drahtgebundener und drahtloser Netzwerktechnologien und kann so den fortschreitenden Entwicklungen im Bereich der Netzwerktechnologien Rechnung tragen.