Lehrstuhl für Kommunikationsnetze
Technische Universität München

Mobilitätsunterstützung mit Programmierbaren Netzen

Peter Tabery

Vollständiger Abdruck der von der Fakultät für
Elektrotechnik und Informationstechnik der Technischen Universität München
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. habil. Alexander W. Koch
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. Jörg Eberspächer
	2.	Univ.-Prof. Dr.-Ing. Adam Wolisz, Technische Universität Berlin

Die Dissertation wurde am 20.06.2006 bei
der Technischen Universität München
eingereicht und durch die
Fakultät für Elektrotechnik und Informationstechnik
am 20.11.2006 angenommen.

Zugleich veröffentlich in der Reihe Kommunikationstechnik (Band 20)

Herbert Utz Verlag GmbH, München
089-277791-00 · www.utz.de

ISBN 978-3-8316-0685-6

Kurzfassung

Neben den klassischen Mobilfunknetzen wie beispielsweise GSM, deren Hauptanwendung die Sprachkommunikation ist, etabliert sich zusehends der drahtlose Internetzugang über Funkzugangsnetze – allen voran Wireless LAN. Während jedoch die Unterstützung der Mobilität des Teilnehmers bei den Mobilfunknetzen per se vorhanden ist, fehlt diese Funktion bei den Funkzugangsnetzen zumeist noch. Für die IP-basierte Kommunikation wurde von der IETF mit Mobile IP ein Standard für beide Aspekte der Mobilität geschaffen, *Roaming* und *Handover*. Außer der reinen Zustellung von Datagrammen für mobile Teilnehmer sind für eine effiziente Kommunikation jedoch auch weitergehende, flexibel bereitstellbare Dienste auf Netzseite erforderlich.

Beim *Roaming* stellen sich dabei zwei Herausforderungen. Einerseits werden dem Teilnehmer nicht überall die gleichen Dienste angeboten, so daß Dienste nach einem Handover in ein anderes Netz unter Umständen überhaupt nicht verfügbar sind. Andererseits kann der Teilnehmer nur auf diejenigen den Handovervorgang unterstützenden Funktionen zurückgreifen, die von beiden Betreibern in gleicher Weise bereitgestellt werden. Diese beiden Herausforderungen lassen sich mit der Technologie der *Programmierbaren Netze* angehen. Diese entstand aus der Idee heraus, dem Teilnehmer die Möglichkeit zu geben eigene netzseitige Dienste zu instantiieren. Im mobilen Umfeld gestattet ihm die Eigenschaft der dynamischen Ladbarkeit, sowohl passende Optimierungsfunktionalitäten für die Paketzustellung als auch maßgeschneiderte Funktionen die an die Eigenschaften des mobilen Teilnehmers angepaßt sind.

In dieser Arbeit wird eine neuartige Architektur für transparente mobile programmierbare Netzdienste beschrieben, die besonders die Aspekte der kontinuierlichen Bereitstellung derartiger Dienste für mobile Teilnehmer berücksichtigt. Die dafür wesentlichen Funktionen (Mobilitätserkennung, -steuerung und -routing) werden dazu vom allgemeinen Dienstmanagement als eigenständige Prozesse getrennt, um eine möglichst große Flexibilität hinsichtlich der Erweiterbarkeit beziehungsweise Austauschbarkeit dieser Funktionalitäten und ihrem jeweiligen Ausführungsort zu erhalten. Zusammen regeln diese drei Mobilitätsfunktionen dann sowohl alle Aspekte der Mobilität der eigentlichen Dienste, als auch – gegebenenfalls in Kooperation mit *Mobile IP* – die datenstromspezifische Zustellung der Datenpakete zum mobilen Teilnehmer.

Für die Mobilitätssteuerung ergibt sich so ein zusätzlicher Freiheitsgrad im Falle eines Handover. Nach einem Handover kann ein Netzdienst an den neuen Ort transferiert oder alternativ eine vorübergehende Tunnelung der Daten eingerichtet werden. Welche der beiden Strategien einen geringeren Aufwand erwarten läßt, entscheidet die Mobilitätssteuerung anhand einer Schätzung des zu erwartenden Aufwandes. Die Evaluierung des dazu vorgeschlagenen Algorithmus erfolgt auf Basis einer abstrahierenden analytischen Untersuchung des Handoververhaltens. Dazu wird die bislang üblicherweise verwendete von den zellularen Mobilfunknetzen stammende Modellierung um den Effekt der Handover-Hysterese erweitert, welcher bei der von Wireless LAN verwendeten Kanalauswahlstrategie deutlich zu Tage tritt.

Die wesentlichen Aspekte eines beispielhaften transparenten mobilen programmierbaren Netzdienstes wurden erfolgreich in einer prototypischen Implementierung evaluiert, die zugleich die Machbarkeit des Ansatzes belegt. Simulative Untersuchungen komplementieren die Ergebnisse.

Abstract

Besides conventional cellular mobile radio networks like, e.g., GSM, with voice communication being the main application, other technologies primarily used for local Internet access – especially Wireless LAN – are gaining momentum. However, while mobility is an inherent feature of cellular radio networks, it is not well supported by the various wireless extensions of local area networks yet. For IP-based communications, the IETF has created Mobile IP, a standard supporting both aspects of mobility, *roaming* and *handovers*. However, besides pure datagram delivery, mobile subscribers need advanced network services which should be provided in a flexible way in order to enable efficient communication.

In a roaming scenario, there are two major challenges. On the one hand, subscribers are not being offered the identical set of services everywhere. Thus, some services may not be available after performing a handover to a foreign access network. On the other hand, the subscriber may only use those functions for enhancing the handover performance, which are being provided by both of the operators, the previous as well as the new one. Both of these challenges may be solved by employing the technology of *Programmable Networks*, whose core aim lies in enabling the subscriber to establish services within the network by himself. In mobile environments, these dynamically loaded services may provide, e. g., particular functions adapted for optimized datagram delivery or functions for mobility support specifically matching the subscribers' characteristics.

In this thesis, a novel architecture for transparent mobile programmable network services is described, that addresses especially the aspects of continously providing such services to mobile subscribers. The essential functions targeting the latter (mobility detection, control and routing) are separated from the general service management as self-contained processes in order to allow for utmost flexibility regarding their extensibility or exchangeability, respectively, as well as their lieu of execution. Together, these three mobility-related functions control all aspects of the underlying services' mobility as well as – in close cooperation with *Mobile IP*, if applicable – the delivery of datagrams to the mobile subscriber, differentiated by the applications' data flows.

An additional degree of freedom for controlling the subscribers' mobility, especially in case of handovers, arises from the flexibility introduced. Following a handover, a network service may either be transferred to the new point of attachment. Alternatively, a transient tunneling of the respective data flows may be initiated. The mobility control selects the most appropriate of these methods by estimating the expectancy of the accumulated effort associated with each of the two. The respective algorithm proposed herein is evaluated using an abstracted analytical examination of the subscribers' handover behavior. For that purpose, the modeling used hitherto for cellular mobile radio networks is extended to encompass the effect of handover hysteresis that appears inevitably with Wireless LANs channel selection strategy.

The core aspects of an exemplary transparent mobile programmable network service have been evaluated successfully in a prototype implementation, proving the viability of the approach at the same time. The results are complemented by means of simulation.

Vorwort

Diese Arbeit entstand im Rahmen meiner Forschungstätigkeit am Lehrstuhl für Kommunikationsnetze der Technischen Universität München.

Besonderer Dank gebührt meinem Doktorvater Prof. Jörg Eberspächer dafür, daß er mir die Möglichkeit gegeben hat an seinem Lehrstuhl zu forschen. Nicht zuletzt war es seine wissenschaftliche Begleitung, die diese Dissertation ermöglicht hat. Bedanken möchte ich mich auch bei Prof. Adam Wolisz für die Übernahme des Zweitgutachtens.

Stellvertretend für alle Kollegen am Lehrstuhl sei namentlich Dr. Christian Hartmann und Dr. Christian Bachmeir gedankt. Der wissenschaftliche Diskurs mit ihnen stellte eine große Bereicherung meiner Arbeit dar und half die Untiefen erfolgreich zu umschiffen. Mein Dank gilt auch Dr. Martin Maier, der stets ein offenes Ohr für meine Hard- und Softwaresorgen hatte.

Herzlichen Dank verdient nicht zuletzt auch meine Lebensgefährtin Birgit Reinisch, die mich während der Entstehung dieser Arbeit liebevoll unterstützt hat und viel Geduld für mich aufbrachte.

München, im Dezember 2006

Inhaltsverzeichnis

Kapitel 1

Einführung

Die Signale über das erste transatlantische Telegraphenkabel im Jahre 1857 [Web99] markieren den Aufbruch in das Zeitalter der globalen Telekommunikation. Von den ersten Telegrammen war es jedoch noch ein weiter Weg bis zur paketorientierten Datenvermittlung, die im Jahr 1969 im damaligen *Arpanet*, dem Vorläufer des heutigen Internet, ihren Anfang fand [LCC+03]. Die zweite Säule der modernen Kommunikationstechnik, die drahtlose Nachrichtenübertragung, begann mit dem experimentellen Nachweis der elektromagnetischen Wellen durch *Heinrich Hertz* im Jahre 1888 [Her88]. Neben den leistungsstarken Sendern, die Nachrichten weltumspannend verbreiten können, entstand später das Konzept des *zellularen Mobilfunks* mit geringen Sendeleistungen und kleinen Reichweiten. Hierbei stellt das Funkmedium nicht direkt die Verbindung vom Sender zum Empfänger her, sondern dient als Zugang zu vermittelnden Festnetzen.

Die heute überwiegend genutzten Funkzugangsnetze lassen sich kategorisieren in Mobilfunknetze mit besonderer Unterstützung für die Sprachübertragung und bloße Datenzugänge. Zu ersteren zählen das *Global System for Mobile Communications* (GSM) [BVE99] und das *Universal Mobile Telecommunications System* (UMTS) [KAL+01], die beide lizenzierte Frequenzbänder nutzen und volle Mobilitätsunterstützung bieten. Die Datenfunksysteme, wie insbesondere das *Wireless LAN* nach dem Standard IEEE 802.11, waren zunächst lediglich als drahtlose Erweiterung der lokalen Datennetze und im unlizenzierten *Industrial, Scientific, and Medical* (ISM) Frequenzband konzipiert. Abzugrenzen davon sind Systeme, die in erster Linie nicht für Infrastrukturzwecke, sondern zur *Ad Hoc*-Kommunikation verschiedener Mobilgeräte gedacht sind, wie beispielsweise IEEE 802.15.1/Bluetooth.

Im Zuge der aktuellen Weiterentwicklung verschwinden die Unterschiede zwischen den Systemen allerdings zunehmend. Einerseits werden bei UMTS die verfügbaren Datenraten erhöht wie mit dem zur Zeit eingeführten *High Speed Downlink Packet Access* (HSDPA) [3GP04]. Andererseits wird mit IEEE 802.11f das Wireless LAN um gewisse Funktionen zur Mikro-Mobilitätsunterstützung erweitert. Mit dem derzeit standardisierten IEEE 802.16/WiMAX entsteht dagegen ein hochbitratiges Funksystem für lizenzierte Funkbänder, welches optional auch Mobilitätsunterstützung bietet. Da im Zuge der *Konvergenz der Dienste* auch über ein Datennetz

vollwertige Sprachdienste mitsamt vielfältiger Leistungsmerkmale – einschließlich *Roaming*- und Handoverunterstützung – angeboten werden können, entwickeln sich die ursprünglich für die Datenübertragung vorgesehenen Systeme zu einer ernsthaften Konkurrenz für die etablierten Mobilfunksysteme und können daher als *disruptive Technologien* im Sinne von [Chr03] bezeichnet werden.

Neben diesen Entwicklungen im Bereich des Funkzuganges erfolgt jedoch auch eine Evolution der dahinterliegenden Netz- und Dienstinfrastruktur. Nach der in der Vergangenheit von statten gegangenen Umstellung der öffentlichen Fernsprechnetze auf das *Intelligente Netz* (IN) erleben wir aktuell mit der Migration zur paketorientierten *Voice over IP*-basierten Sprachübertragung den nächsten großen Umbruch in der Welt der Telekommunikation. Damit einher geht auch eine Umstellung der Dienststeuerung von den Zeichengabesystemen der ITU (SS7 [Int93c], ISUP [Int99], etc.) auf das von der IETF spezifizierte *Session Initiation Protocol* (SIP) [RSC+02]. Das Konzept des *Softswitch* [OJ03, Ebe06] führt dabei aufgrund der Trennung von Dienstlogik und Vermittlungsvorgang zu einer weiteren Vereinfachung der Entwicklung neuer Telekommunikationsdienste.

Neben der Kostenersparnis ist auch die zusätzliche *Flexibilität* bei der Bereitstellung innovativer Leistungsmerkmale und Kommunikationsdienste ein wesentlicher Faktor bei der Umstellung der Systeme. Mit dem drastischen Rückgang der Preise für Kommunikationsdienstleistungen in den vergangenen Jahren [Lin06] sind die Betreiber gezwungen, ihre Netze noch kosteneffizienter zu betreiben und rasch von den Kunden gewünschte neuartige Dienste anzubieten, um sich von ihren Konkurrenten abheben zu können.

Einen neuartigen Ansatz zur dynamischen Erweiterung der Funktionalität von Netzknoten stellen die *Aktiven und Programmierbaren Netze* [CDK+99] dar. Diese gestatten es, strukturiert das Ausführen von Programmfunktionen auf dedizierten Netzknoten von der Ferne aus zu veranlassen, wobei das Ziel die Bearbeitung von Datenströmen eines Teilnehmers ist. Die Funktionen gehen typischerweise über das reguläre Routing von Datagrammen hinaus und können Elemente verschiedener Protokollfunktionen umfassen.

1.1 Problemstellung und Lösungsansatz der Arbeit

In der vorliegenden Arbeit wird ein *Wireless LAN*-Netz mit mehreren *Access Points* angenommen, durch das sich ein mobiles Terminal frei bewegt. Im besten Falle kann das Terminal mit mehreren Basisstationen gleichzeitig kommunizieren und so stets eine funktionierende Verbindung mit dem Internet haben, selbst wenn der Funkkontakt zu einer Basisstation plötzlich abbrechen sollte. Typischerweise sind heutige Wireless LAN-Adapter aber nur in der Lage, sich mit einer einzigen Basisstation gleichzeitig zu assoziieren. Daher verursacht ein Wechsel der Basisstation immer auch eine Unterbrechung der Konnektivität. Dies ist immer mit Paketverlusten verbunden, gegen die Wireless LAN nicht in der Lage ist, Vorkehrungen zu treffen.

Ferner wird davon ausgegangen, daß der Teilnehmer für seine Kommunikation mit entfernten Teilnehmern auch auf *netzseitige Dienste* zurückgreift. Diese werden auf dedizierten Knoten im

Zugangsnetz ausgeführt und unterstützen die Anwendungen des Teilnehmers in spezifischer Art und Weise. Da diesen Anwendungen verschiedenste Protokolle zugrunde liegen und der Dienst womöglich für verschiedene Terminals in unterschiedlicher Weise erbracht werden soll, bietet es sich an, sich der Flexibilität der programmierbaren Dienste zu bedienen. Diese ermöglichen es, flexibel eine Vielzahl von Diensten ohne vorherige explizite Installation zur Verfügung zu stellen. Dies ist insbesondere beim *Roaming* der Terminals von Vorteil, da diese sich dabei mit fremden Zugangsnetzen verbinden, in denen der erforderliche Dienst nicht notwendigerweise vorab vorhanden ist. Daher stellt es eine pragmatische Lösung dar, standardisierte Softwarekomponenten bei Bedarf aus einem zentralen Speicher zu laden und dynamisch zu den gewünschten Diensten zu kombinieren.

Die Mobilität des Teilnehmers bedingt jedoch auch, daß die Dienste nach einem *Handover* des Teilnehmers am neuen Aufenthaltsort zügig bereitgestellt werden. Dabei sollte der Transfer des Dienstes die durch den Handover verursachte Unterbrechung der Datenübermittlung nicht wesentlich verlängern. Gleichzeitig ist jedoch darauf zu achten, daß jeder Diensttransfer Netzressourcen beansprucht, wodurch möglicherweise die Dienste anderer Teilnehmer beeinträchtigt werden.

In dieser Arbeit wird eine Architektur entworfen, welche die Bereitstellung eines mobilen programmierbaren Dienstes sowohl durch *Diensttransfer* als auch durch *Datentunnelung* ermöglicht. Diese macht sich die Möglichkeiten der Programmierbaren Netze zu nutze, indem auch die Steuerungsfunktionen als programmierbare Dienste realisierbar sind. Die Architektur unterscheidet dabei drei Ebenen, nämlich das Management zur Koordinierung aller Dienste eines Teilnehmers, die Dienststeuerung für die Elemente eines Dienstes und schließlich die darunterliegende ausführende Ebene. Dabei wird der mobilitätsspezifische Teil vom Management abgespalten, um eine schnellere Reaktion auf Handover zu ermöglichen. Der modulare Aufbau gestattet eine einfache Anpassung an die spezifischen Gegebenheiten. Dies wird exemplarisch anhand der Implementierung des *Programmierbaren Proxy* als programmierbarer Dienst gezeigt.

Die darunterliegende programmierbare Plattform wird dabei weitestgehend abstrahiert betrachtet. Es wird lediglich gefordert, daß diese die Ausführung transparenter Dienste – also ohne Zutun der Anwendungsprogramme des Teilnehmers – gestattet. Dies bedeutet, daß die Dienstsignalisierung vom Anwendungsdatenstrom abgekoppelt wird.

Da die entsprechenden Managementfunktionen mit den passenden Schnittstellen *dynamisch* geladen werden können, ist die Architektur unabhängig von den verwandten Verfahren zur Mobilitätsunterstützung. Wird beispielsweise das *Mobile IP* zur Makro-Mobilitätsunterstützung verwendet, kann die Mobilitätsmanagementeinheit auf die dort vorhandene Erkennung des Teilnehmers in einem neuen Netz zurückgreifen. Sie muß jedoch selbst eine Signalisierung in das zuvor genutzte Zugangsnetz zu den dort genutzten Diensten durchführen, da dies regulär nicht durch Mobile IP erfolgt. Die Dienststeuerung jedes Dienstes wird dann entweder einen Transfer desselbigen durchführen oder eine Tunnelung des betreffenden Anwendungsdatenstromes einleiten.

Für die Entscheidung, ob ein Diensttransfer oder eine Datentunnelung den geringeren Aufwand erwarten läßt, wird ein einfacher *Algorithmus* entworfen, der keine Angaben über die Mobilität des einzelnen Teilnehmers verwendet. Dieser trifft seine Entscheidung auf Grundlage mehrerer

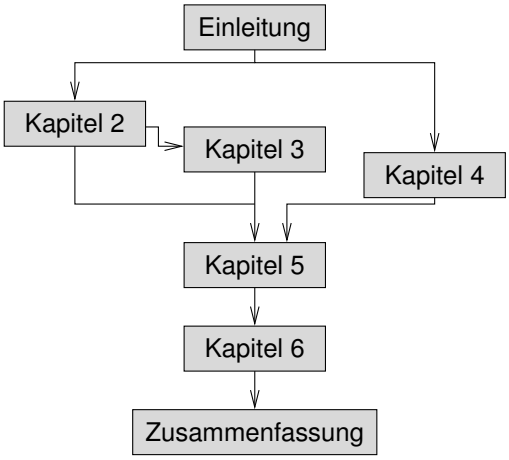


Abbildung 1.1: Logischer Fluß der Gliederung dieser Arbeit.

Kriterien. Erstens ist dies der Aufwand für den Transfer der einzelnen Dienstinstanz, zweitens die Datenrate der diesen Dienst nutzenden Anwendung. Schließlich wird anhand der zu erwartenden Dauer bis zum nächsten Handover die Kostenmetrik für den Diensttransfer sowie die Datentunnelung berechnet.

Die Zwischenhandoverzeit läßt sich für den Einzelfall nie genau vorhersagen. Es ist jedoch möglich, auf der Grundlage einer analytischen Modellierung der Eigenschaften des Wireless LAN mittels statistischer Berechnungen eine Wahrscheinlichkeitsdichtefunktion aufzustellen. Die für herkömmliche Zellularnetze verwendeten Methoden beinhalten jedoch nicht die Eigenschaften der Kanalwahl beim Wireless LAN. In dem hier vorgestellten genaueren Berechnungsverfahren wird der bei Wireless LAN entstehende *Hysterese-Effekt* beim Handover mitberücksichtigt, der nicht nur durch die Eigenschaften der Schwellwerte, sondern auch durch geometrische Gegebenheiten bedingt ist.

1.2 Struktur und Gliederung

Die beiden Fundamente dieser Arbeit sind die Mobilität und die Technologie der Programmierbaren Netze. Beide erfahren eine ausführliche Würdigung in einem eigenen Kapitel. Erst in Kapitel 5 findet mit der dort vorgestellten Architektur eine Verknüpfung dieser beiden Aspekte statt. Der daraus resultierende logische Fluß ist in Abb. 1.1 schematisch dargestellt.

In Kapitel 2 werden die Ansätze zur *Mobilitätsunterstützung* für den Bereich der Mobilität im Internet vorgestellt sowie Gemeinsamkeiten und Unterschiede zu den zellularen Mobilfunknetzen aufgezeigt. Dabei wird jeweils dargelegt, wie Mobilität in den verschiedenen Schichten unterstützt werden kann und wodurch sich die einzelnen Lösungen hinsichtlich Funktionsweise und Wirkung unterscheiden.

Kapitel 3 widmet sich den Eigenschaften von Wireless LAN und dessen Kanalselektionsalgorithmus aus Systemsicht. Daraus wird dann die für die Performanz des Mobilitätsmanagements

wichtige Verteilung der Zwischenhandoverzeit für eine idealisierte Topologie abgeleitet. Im Gegensatz zur bislang üblichen Modellierung der bloßen Zellaufenthaltsdauer berücksichtigt die *Zwischenhandoverzeit* auch die auftretende Hysterese.

In Kapitel 4 wird zunächst ein Überblick über Telekommunikationsdienste gegeben und der Begriff des transparenten programmierbaren Netzdienstes konkretisiert. Anschließend wird auf Dienste in *Aktiven und Programmierbaren Netzen* eingegangen. Nach einem vergleichenden Überblick über die verschiedenen Knotenarchitekturen werden einige Dienste exemplarisch dargelegt. Schließlich wird ein abstrahiertes Modell einer programmierbaren Plattform vorgestellt.

In Kapitel 5 werden die Eigenschaften und Anforderungen *transparenter programmierbarer Netzdienste* im mobilen Umfeld analysiert. Daraus wird dann eine flexible, programmierbare Architektur entwickelt, bei der programmierbare Plattform und Mobile IP Foreign Agent gekoppelt werden. Anschließend wird aufgezeigt, wie diese Architektur die Bereitstellung mobiler Dienste unterstützt und ein *Algorithmus zur Handoveroptimierung* mobiler Dienste vorgestellt und evaluiert.

In Kapitel 6 wird der *Programmierbare Proxy* als Beispiel eines programmierbaren Dienstes vorgestellt und anhand einer prototypischen Implementierung evaluiert. Dabei werden die für diesen Anwendungsfall wesentlichen Aspekte der entworfenen Architektur aufgegriffen.

Die Nomenklatur dieser Arbeit orientiert sich weitgehend an der *Internet Engineering Task Force* (IETF), die deutschen Begriffe entsprechen den Empfehlungen des VDE [EFH$^+$96a, EFH$^+$96b]. Die Begriffe “mobiler Teilnehmer”, “mobiles Terminal” und “mobiles Endgerät” werden in der vorliegenden Arbeit wie Synonyme verwandt, ebenfalls “Basisstation”, “Access Point” und “Netzzugangspunkt”.

Kapitel 2

Mobilitätsunterstützung im Internet

Als um das Jahr 1980 die heutzutage immer noch überwiegend verwendete *vierte* Version des Internet Protokolls (IPv4) [Pos81] konzipiert wurde, war allein aufgrund der Größe der Endgeräte nicht an Mobilität zu denken. Daraus resultierte ein rein auf den Fall weitgehend ortsfester Terminals beschränktes Protokoll. Diese damalige Grundannahme hat ihre Gültigkeit mittlerweile verloren, sind doch tragbare und dabei leistungsfähige Computer heutzutage weit verbreitet. Für die uneingeschränkte Kommunikation mit einem mobilen Teilnehmer sind die in [Ebe02] genannten Kriterien zu erfüllen:

Lokalisierung: Finden beziehungsweise Verfolgen des Teilnehmers

Verbindungsmanagement: Etablieren und Aufrechterhalten einer oder mehrerer Nutzkanal- und Signalisierungsverbindungen

Diese Merkmale werden auch mit den Begriffen *aktive* und *passive Konnektivität* sowie *Handoverunterstützung* bezeichnet. Ein erster Schritt hin zur Unterstützung mobiler Teilnehmer war das Ermöglichen von *Portabilität* mittels BOOTP [CG85] und DHCP [Dro97]. Diese Protokolle sind in der Lage, einem Terminal automatisch eine IP-Adresse zuzuweisen, so daß dieses kommunizieren kann (aktive Konnektivität). Damit wurde das Problem der manuellen Vergabe [vdHKvM98] und Konfiguration von IP-Adressen gelöst.

Ein nächster Schritt war die Einführung von *dynamischem DNS* [VTRB97], welches eine einfache, aber wenig elegante Lösung für das Problem der *passiven Konnektivität* (Erreichbarkeit) eines mobilen Teilnehmers mit wechselnder IP-Adresse ist. Hierbei wird die Zuordnung von Rechnername und IP-Adresse stets aktualisiert, so daß der Teilnehmer unter der geänderten Adresse erreicht werden kann. Allerdings ist die Gültigkeit einer Zuordnung zeit- und nicht ereignisgesteuert. Wählt man eine sehr kurze Gültigkeitsdauer (Speicherzeit 1 s oder sogar 0 s, was bedeutet, daß die Zuordnung nur einmalig verwendet und nicht gespeichert wird [Moc87]), werden Änderungen der Netzadresse natürlich rasch verbreitet. Allerdings nimmt im selben Maße das Aufkommen von DNS-Anfragen zu, selbst wenn der Teilnehmer seinen Aufenthaltsort nicht ändert.

Tabelle 2.1: Grad der Mobilitätsunterstützung im Vergleich.

	Feste (manuelle) Konfiguration	DHCP	DHCP mit dynamischem DNS	Mobile IP
Aktive globale Konnektivität	–	✓	✓	✓
Passive globale Konnektivität	–	–	✓	✓
Unterstützung von Transportverbindungen bei Netzwechseln	–	–	–	✓
Endgerät ist:	**Ortsfest**	**Portabel**	**Portabel**	**Mobil**

Um die Unzulänglichkeiten dieser Ansätze zu überwinden, wurde von der IETF das *Mobile IP* [Per02] spezifiziert. Dieses Protokoll erfüllt alle oben genannten Anforderungen an die Unterstützung von Mobilität, wie im Vergleich in Tabelle 2.1 gezeigt wird. Bedingt durch die Tatsache, daß es sich dabei um eine reine *Netzschichtlösung* handelt, ist jedoch noch großes Verbesserungspotential vorhanden. In Mobile IP flossen die Erfahrungen aus verschiedenen Ansätzen akademischer und industrieller Forschung ein, so daß es die Obermenge der bis dato existierenden Vorschläge zur Mobilitätsunterstützung auf der Netzschicht darstellt [BPT96]. Daher beschränkt sich diese Arbeit auf die Betrachtung von Mobile IP.

Andererseits bieten heutige zellulare Mobilfunknetze (die selbstverständlich über eine ausgereifte Mobilitätsunterstützung verfügen) auch Datendienste. Beim *Global System for Mobile Communications* (GSM) [EVB01], dem mit Abstand bedeutendsten System der sogenannten zweiten Generation des Mobilfunks, wurden beispielsweise *High Speed Circuit-Switched Data* (HSCSD) und *General Packet Radio Service* (GPRS) nachträglich integriert (weshalb man hierbei auch von der 2,5-ten Generation spricht). Durch die Verwendung neuer Modulationsverfahren (*Enhanced Data Rate for GSM Evolution*, EDGE) konnte die maximale Datenrate weiter gesteigert werden. Bei den derzeit im Aufbau befindlichen Systemen der dritten Generation (insbesondere das *Universal Mobile Telecommunications System*, UMTS) sind Datendienste hingegen ein integraler Bestandteil, zugleich liegen die erzielbaren Datenraten deutlich höher (in etwa um eine Größenordnung).

Gegenüber *Wireless LAN* nach IEEE 802.11 sind die Datendienste zellularer Mobilfunknetze durch geringere Bitraten, höhere Latenzzeiten und höhere Kosten gekennzeichnet. Sie bieten jedoch dafür eine örtliche Verfügbarkeit, wie sie mit dem unlizenzierten und daher notwendigerweise unkoordinierten Wireless LAN nicht erreicht werden kann.

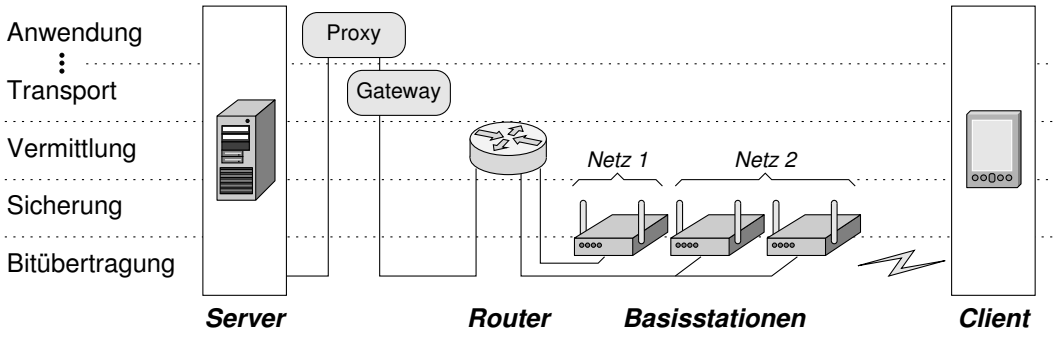


Abbildung 2.1: Netzelemente im Schichtenmodell aus Sicht eines mobilen Clients.

Um die Vorteile beider Systeme verbinden zu können, muß jedoch die Leistungsfähigkeit der Mobilitätsunterstützung bei der Verwendung von Internetzugangstechniken, wie beispielsweise dem Wireless LAN, verbessert werden. Ansatzpunkte bieten sich dabei in den verschiedenen Schichten des ISO/OSI-Schichtenmodells.

2.1 Klassifizierung der Mobilitätsunterstützung

2.1.1 Terminalmobilität im Schichtenmodell

Die Einordnung der verschiedenen Netzelemente in das ISO/OSI-Schichtenmodell [Int93b] aus Sicht eines mobilen Clients ist in Abb. 2.1 dargestellt[1]. Auf der *Bitübertragungsschicht* ist die Mobilität des Terminals auf den von der aktuellen Funkzelle abgedeckten Bereich beschränkt. Durch die Verwendung von Repeatern kann dieser Bereich erweitert werden.

Anders sieht es bei der Mobilitätsunterstützung auf der *Sicherungsschicht* aus. Hier besitzen die Basisstationen zugleich die Funktionalität einer *Bridge*. Auf diese Weise kann der mobile Teilnehmer im Bereich aller Basisstationen des aktuellen IP-Netzes kommunizieren. Wechselt er aus dem Bereich einer Basisstation in den einer anderen, beginnt diese mit der Weiterleitung der an ihn adressierten Rahmen. Die vorherige Basisstation wird die Zustellung der Datagramme nach einer bestimmten Zeit einstellen.

Diese einfache Art der Mobilitätsunterstützung der Sicherungsschicht funktioniert nur, da hier eine flache Adressierung ohne Topologieinformationen verwendet wird. Anders sieht es aus, wenn ein mobiles Endgerät in den Bereich einer Basisstation wechselt, die an ein anderes IP-Netz angeschlossen ist. Da die IP-Adresse zugleich die Adresse des IP-Netzes beinhaltet, in dem sich der Teilnehmer aufhält (beziehungsweise aufhalten sollte), muß eine Mobilitätsunterstützung auf der *Vermittlungsschicht* das reguläre IP-Routing aktiv ändern oder umgehen.

[1] Aus Gründen der Übersichtlichkeit wurden die Kommunikationssteuerungs- und Datendarstellungsschicht ausgelassen. Sie sind im Bereich der Netze ohnehin nur von untergeordneter Bedeutung.

Dieses Problem wird vermieden, wenn die Mobilitätsunterstützung *oberhalb* der Vermittlungsschicht angesiedelt wird. Wird dem Kommunikationspartner (Server) nach dem Wechsel des IP-Netzes die neue IP-Adresse mitgeteilt, kann jener die *Transportverbindung* mit der neuen IP-Adresse assoziieren. Alternativ kann dies auch die *Anwendung* selbst ausführen, vorausgesetzt sie verfügt über eigene Mechanismen zur Datensicherung, da die der Transportschicht dann außer Kraft gesetzt werden.

Man beachte jedoch, daß die Dauer eines Handover maßgeblich von der Paketumlaufzeit zwischen den Umschaltpunkten bestimmt wird. Befinden sich diese im Client und Server, kann die dadurch bedingte Verzögerung deutlich länger sein als im Fall des Umschaltens in dazwischenliegenden Netzelementen.

2.1.2 Mikro- und Makro-Mobilität

Je nachdem, ob ein Verfahren zur Mobilitätsunterstützung global oder nur örtlich beschränkt funktioniert, wird es als *makro-* oder *mikro*-mobilitätsunterstützend kategorisiert. Der Wirkungsbereich von Mikro-Mobilitätsunterstützung wie beispielsweise *Cellular IP* [Val99] oder *Hawaii* [RLTV99] ist immer auf die (Funk-)Netze einer administrativen Einheit und einer Technologie beschränkt. Die Vorteile lassen sich also nur bei solchen Standortwechseln nutzen, die zwischen benachbarten Einheiten erfolgen die einer Kontrollinstanz unterstehen. Dann erfolgt die Reaktion auf einen Standortwechsel vor Ort, so daß insbesondere keine Latenzen entstehen wie sie zwangsläufig mit einer externen Signalisierung verbunden sind.

Im Gegensatz dazu sind Makro-Mobilitätsprotokolle wie Mobile IP [Per02] dadurch gekennzeichnet, daß auch heterogene Netzwechsel unterstützt werden, also zwischen administrativen Domänen und unterschiedlichen Technologien. Zu diesem Zweck sind Makro-Mobilitätsprotokolle so gehalten, daß sie von der Sicherungsschicht unabhängig sind und auch keine besondere Zustandshaltung vor Ort (z. B. Nachbarschaftsinformation) verlangen. Beides verringert jedoch auch die Geschwindigkeit mit der auf Netzwechsel reagiert werden kann.

Um in den Genuß der Vorzüge beider Verfahren zu kommen, können diese kombiniert werden, wie in Abb. 2.2 schematisch gezeigt. Einfach möglich ist das bei Mikro-Mobilitätsprotokollen, die allein auf der Sicherungsschicht arbeiten und somit auf ein IP-Netz beschränkt sind, wie beispielsweise [Val99]. Diese sind für Mobile IP üblicherweise voll transparent, das heißt, daß weder eine Anpassung der Bewegungserkennung noch der Signalisierungsnachrichten erforderlich ist.

Deutlich schwieriger ist die Integration bei solchen Mikro-Mobilitätsprotokollen, die sich über mehrere IP-Netze erstrecken und dabei auf Funktionalität des IP-Routing aufbauen, wie dies beispielsweise bei [RLTV99] oder [FWW02] der Fall ist. In diesem Sonderfall müssen diese Protokolle bis ins Detail mit Mobile IP abgestimmt werden, da auch Mobile IP die Bewegung zwischen IP-Netzen erkennt und dann entsprechende Maßnahmen ergreift, die denen des Mikro-Mobilitätsprotokolls zuwiderlaufen. Einerseits muß die Bewegung innerhalb des Mikro-Mobilitätsbereichs also vor Mobile IP verborgen werden, andererseits müssen aber auch (unter Umständen angepaßte) Signalisierungsnachrichten und Nutzdaten korrekt zugestellt werden.

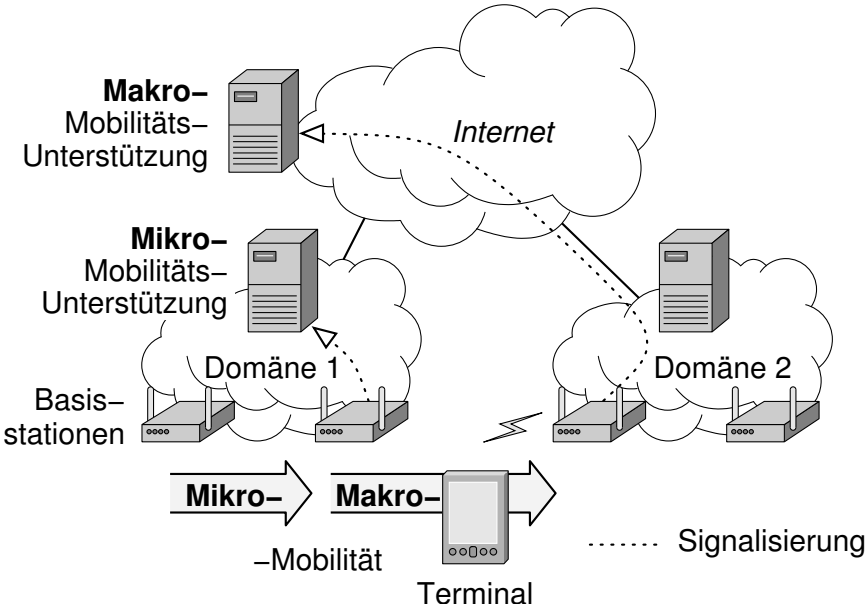


Abbildung 2.2: Signalisierung bei Mikro- und Makro-Mobilitätsunterstützung

Beispielsweise beruht die Erkennung des aktuellen IP-Netzes bei Mobile IP [Per02] auf mittels lokalem Broadcast (TTL=1) versendeten ICMP-Nachrichten, die natürlich nicht ohne weitere Maßnahmen in benachbarte Netze geflutet werden können.

2.1.3 Overlaynetze: Horizontale und vertikale IP-Netzwechsel

Allgemein wird eine Überlagerung verschiedener physischer oder virtueller Netze als Overlaynetz bezeichnet. Virtuelle Netze bieten auf Basis einer physischen Infrastruktur eine abstrahierte Topologie an, wie z. B. ein *Virtual Private Routed Network* (VPRN) [GLH+00].

Bei GSM werden Mikro- und Piko-Zellen dafür eingesetzt, die Netzkapazität an Orten mit hoher Verkehrsdichte punktuell zu verbessern [Lag97]. Die sich ergebende mehrschichtige Overlaynetzstruktur der Zellen ist in Abb. 2.3 schematisch gezeigt. Aus den zur Verfügung stehenden Basisstationen wird dann nicht unbedingt die mit dem höchsten Empfangspegel ausgewählt, sondern diejenige die der Bewegungscharakteristik des Teilnehmers entspricht. Idealerweise werden ruhende Teilnehmer von einer Piko-Zelle bedient, während die Zahl der Handover für sich schnell bewegende Teilnehmer gering gehalten wird, indem diese der Makro-Zelle zugewiesen werden [IS95].

Aus Sicht eines IP-Endgerätes mit mehreren Funkschnittstellen stellen die unterschiedlichen Internetzugangstechniken die verschiedenen Overlayschichten dar, wie dies in Abb. 2.4 dargestellt [SK98] ist. Diese unterscheiden sich typischerweise sowohl hinsichtlich ihrer maximalen Reichweite, der erreichbaren Datenrate, der Paketverzögerung, als auch der mit der Übertragung entstehenden Kosten.

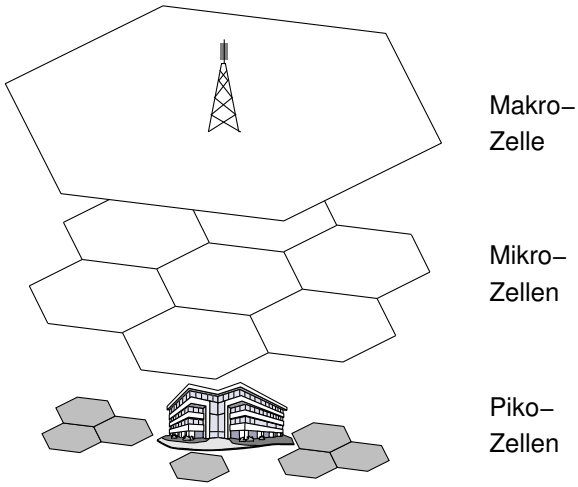


Abbildung 2.3: Mehrschichtige Overlay-Netzstruktur bei GSM.

Wechselt ein Teilnehmer zwischen zwei Basisstationen der gleichen Technologie[2], wird dies als *horizontaler Handover* bezeichnet. Dieser zeichnet sich dadurch aus, daß das mobile Terminal seine IP-Adresse nur dann ändern muß, wenn diese Basisstationen in unterschiedlichen IP-Netzen liegen. Findet der Handover zwischen zwei Basisstationen desselben Anbieters statt, sind diese meist auch im Sinne der Netztopologie nicht weit voneinander entfernt, so daß sich die Auswirkungen des Handover durch Mikro-Mobilitätsprotokolle verringern lassen. Es kann aber wie beispielsweise bei Wireless LAN eine (hardwarespezifische) Unterbrechung der Datenübertragung allein dadurch verursacht werden, daß das Terminal während der Suche nach einem neuen Netz die Kommunikation mit dem bisherigen nicht aufrechterhalten kann (*break-before-make* Handover).

Sind mehrere Netze verfügbar, so kann ein Teilnehmer ein seinen Anforderungen entsprechendes Netz auswählen. Wechselt der Teilnehmer zwischen IP-Netzen mit verschiedenen Zugangstechnologien, bezeichnet man das als *vertikalen Handover*.

Sowohl beim horizontalen wie beim vertikalen Handover läßt sich der Paketverlust reduzieren oder sogar vermeiden, wenn zwei passende Netzschnittstellen zur Verfügung stehen. In diesem Falle kann das Terminal zuerst die Registrierung mit dem neuen Netz initiieren und erst danach die Assoziation mit dem bisherigen Netzzugangspunkt lösen, was auch als ein *make-before-break* Handover bezeichnet wird. Im Falle verschiedener Zugangstechnologien ist es auch möglich, beide Zugangstechnologien parallel für verschiedene Anwendungen zu nutzen, z. B. mit Hilfe des *Mobile Policy Table* [ZCB01]. Im extremsten Fall dient die schmalbandige Zugangstechnologie (beispielsweise GPRS) nur der Übertragung von Steuerungsinformationen [**ZTB03**].

Verfügt das Terminal nur über eine Netzschnittstelle, die sich gegebenenfalls auf eine andere Technologie dynamisch umkonfigurieren läßt (z. B. mittels sogenanntem *Software Defined Ra-*

[2]Im folgenden wird vereinfachend angenommen, daß eine Technologie immer nur die Übertragung von IP-Paketen zu *einem* IP-Netz gleichzeitig zuläßt, wie dies beispielsweise bei GPRS und auch Wireless LAN der Fall ist (vgl. Abschnitt 3.1).

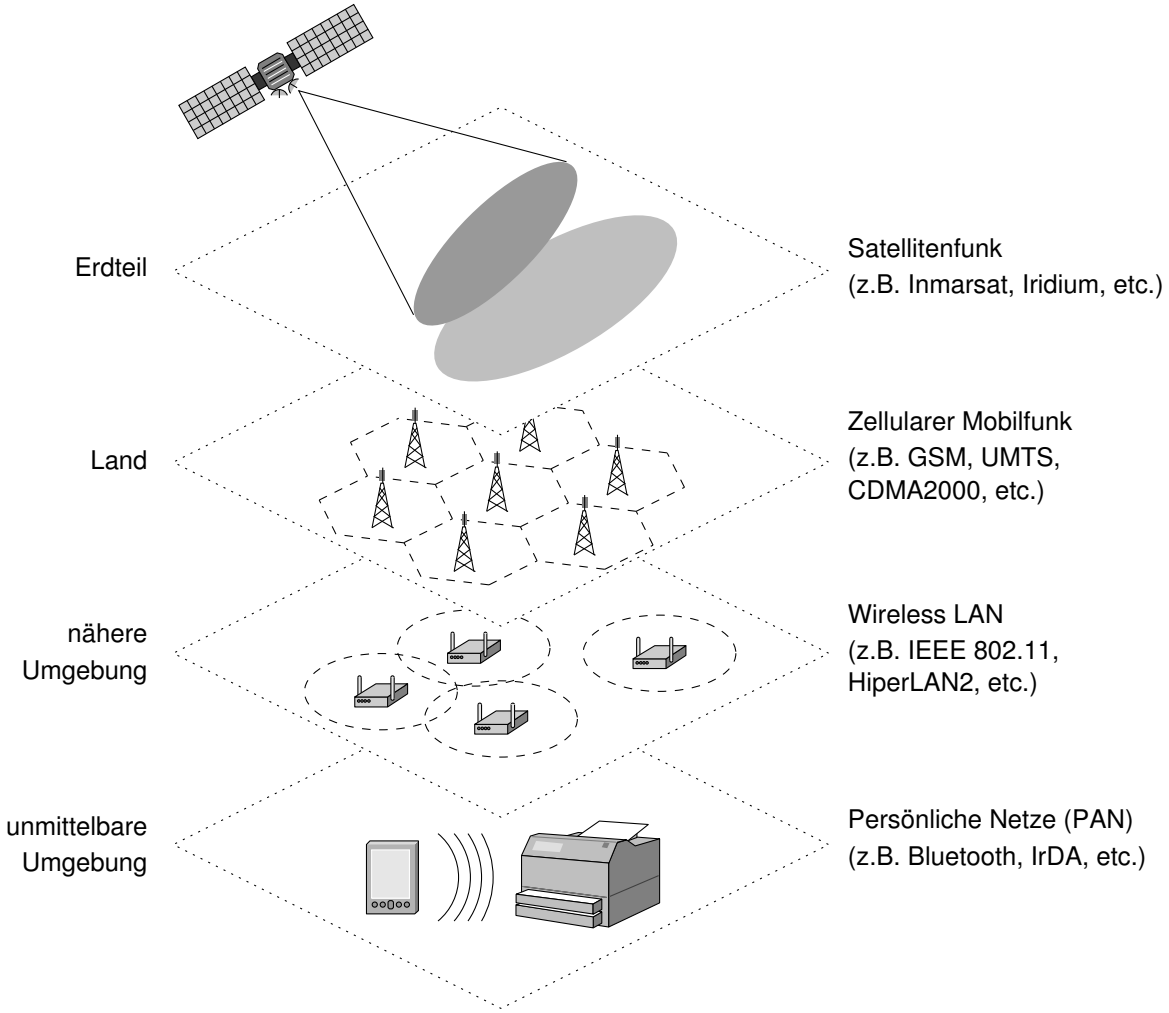


Abbildung 2.4: Drahtlose Overlaynetze aus Sicht eines IP-Endgerätes.

dio [Mit95]), so muß zuerst die Assoziation mit dem aktuellen Netzzugangspunkt abgebrochen werden bevor die neue Registrierung erfolgen kann (*break-before-make* Handover, vgl. oben).

2.1.4 Mobile Drahtlose Ad Hoc-Netze

Im Gegensatz zu Netzen mit festen Basisstationen, bei denen ein mobiles Terminal stets direkt mit einer festen Netzinfrastruktur kommuniziert, stehen mobile drahtlose *Ad Hoc*-Netze. Obgleich dieser Begriff mit unterschiedlicher Bedeutung verwendet wird, so sind es doch zwei Eigenschaften, die dabei stets angenommen werden: Einerseits die Möglichkeit zur Kommunikation ohne feste Infrastruktur, wobei Datenpakete von dazwischenliegenden Terminals empfangen und dann weitergesendet werden (*Multihop*-Kommunikation), andererseits die dezentrale Organisation dieses Vorgangs [HGJ+99].

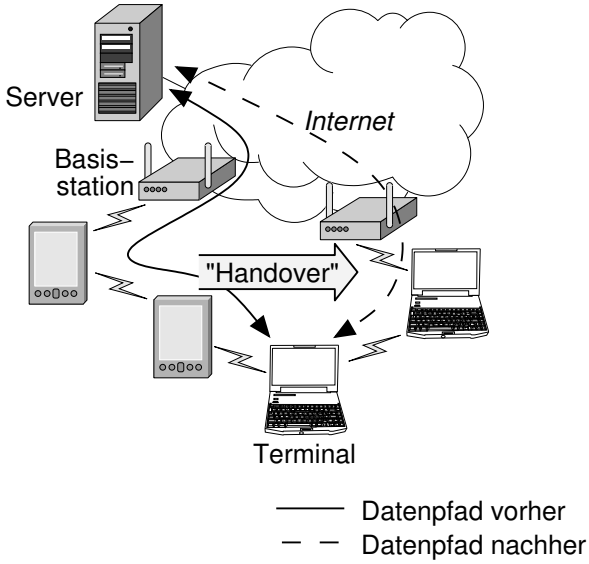


Abbildung 2.5: Nachbildung eines Handover im Fall eines hybriden *Ad Hoc*-Netzes [XB02].

Mobilität der Teilnehmer ist eine in diesem Zusammenhang zwar nicht notwendige, jedoch üblicherweise angenommene Eigenschaft. Durch die Mobilität sowie das Ein- und Ausschalten von Terminals ergeben sich deutlich häufigere Änderungen in der Topologie des Netzes als dies in fest installierten Netzen, wie beispielsweise dem Internet, der Fall ist. Üblicherweise sind also für mobile drahtlose *Ad Hoc*-Netze auch schnell reagierende Routingprotokolle erforderlich, um der sich ändernden Netztopologie Rechnung zu tragen.

Folglich gibt es bei *Ad Hoc*-Netzen auch keinen eindeutigen Handover wie in den weiter oben besprochenen zellularen Systemen. Nutzt man nun in einem hybriden Ansatz ein *Ad Hoc*-Netz, um eine feste Basisstation zu erreichen [XB02], muß der Handovervorgang entsprechend nachgebildet werden, wie in Abb. 2.5 beispielhaft gezeigt. Dadurch läßt sich vermeiden, daß ein mobiler Teilnehmer eine weiter entfernte Basisstation verwendet und so das *Ad Hoc*-Netz unnötig belastet. Für diese Arbeit wird angenommen, daß das darunterliegende Protokoll zur Verwaltung des *Ad Hoc*-Netzes einen Handover so nachbildet, daß die Makro-Mobilitätsunterstützung auf IP-Ebene oder darüber diesen wie einen regulären *make-before-break* Handover behandeln kann.

2.2 Mobile IP

Auf der Netzschicht gibt es zwei fundamentale Konzepte, um Datagramme an ein nicht ortsfestes Endgerät zuzustellen. Bei einem *netzorientierten* Ansatz geht man davon aus, daß die Ortsinformation in weiten Teilen des Netzes so verteilt wird, daß die Pakete stets auf dem besten (weil kürzesten) Weg zum beweglichen Terminal zugestellt werden. Da diese Methode in größeren Netzen mit vielen mobilen Teilnehmern schnell an ihre Grenzen stößt, wird sie im Internet nicht für die Makro-Mobilitätsunterstützung verwandt. Die Verwendung von quellengesteuerter

Verkehrslenkung (*Source Routing*) könnte das Problem der Zustandshaltung in den Netzknoten vermeiden. Da *Source Routing* jedoch inhärent unsicher ist [Per00] und dementsprechend kaum Verbreitung gefunden hat, stellt es keine wirkliche Alternative dar.

Dem gegenüber stehen *endgeräteorientierte* Herangehensweisen, deren bekannteste das von der IETF spezifizierte Mobile IP ist (für IPv4 in [Per02], für IPv6 in [JPA04]). Da hierbei die zur Mobilitätsunterstützung erforderliche Logik in einer einzigen Einheit am Rande des Netzes konzentriert ist (dem sogenannten Home Agent), ist die Einsetzbarkeit weitestgehend unabhängig von der Größe des Gesamtnetzes. Dadurch bedingt nehmen die Datagramme allerdings zumeist nicht den kürzesten Weg vom Sender zum Empfänger.

Dreh- und Angelpunkt bei Mobile IP ist der bereits erwähnte *Home Agent* (HA). Er befindet sich im Heimatnetz des mobilen Teilnehmers, also dem Netz das dessen permanenter IP-Adresse entspricht. Hält sich der mobile Teilnehmer selbst nicht dort auf, registriert er seine aktuelle Erreichbarkeit beim Home Agent. Dieser nimmt dann alle an das mobile Terminal adressierten IP-Pakete entgegen und leitet sie an den durch eine temporäre Zustelladresse (*Care-of Address*) definierten aktuellen Aufenthaltsort weiter.

Dazu werden die zuzustellenden IP-Pakete als Nutzdaten in ein neues IP-Paket *enkapsuliert*, das heißt es wird ein neuer IP-Header vorangestellt. Da das innere IP-Paket für die anderen Netzelemente verborgen ist, werden diese IP-Pakete nun zu der im äußeren, hinzugefügten Header vorhandenen IP-Adresse durch das Internet befördert. Bevor sie das entsprechende Anwendungsprogramm erreichen, werden die Pakete wieder ausgepackt, indem der zusätzliche Header entfernt wird. Da die ursprünglichen IP-Pakete bei diesem Vorgang nicht verändert werden[3], wird dieses Verfahren auch als IP-*Tunnelung* bezeichnet. Vom mobilen Teilnehmer gesendete Datagramme werden zumeist auch durch den Tunnel gelenkt, denn nur dann durchlaufen sie eine eventuell vorhandene *Firewall* mit topologisch korrekten Adressen. Andernfalls könnten sie bei einer Plausibilitätsprüfung in einer dazwischenliegenden Firewall aufgrund ihrer topologisch falschen Absenderadresse herausgefiltert werden [FS00, Mon01].

Meldet sich der mobile Knoten an einem vorhandenen *Foreign Agent* (FA) an, endet der IP-in-IP-Tunnel dort, wie in Abb. 2.6 dargestellt. In diesem Falle ist die temporäre Zustelladresse (*Care-of Address*) die IP-Adresse des Foreign Agent, der dann auch die vom Home Agent eingekapselten IP-Pakete wieder auspackt. Anschließend werden die Pakete (unter Umgehung der regulären Routingmechanismen) direkt an die *MAC-Adresse* des mobilen Teilnehmers zugestellt. Durch die Verwendung eines Foreign Agent benötigen die fremden mobilen Terminals keine eigene temporäre IP-Adresse, was bei der gegenwärtigen Knappheit von IPv4-Adressen durchaus von Vorteil ist.

Andererseits kann der mobile Teilnehmer auch selbst Tunnelendpunkt sein. Dafür benötigt er jedoch eine temporäre IP-Adresse aus dem Adressbereich des besuchten Netzes, die er beispielsweise über DHCP [Dro97] beziehen kann. Da diese temporäre IP-Adresse zusätzlich zur permanenten Heimatadresse verwendet wird, wird sie als *Co-located Care-of Address* bezeichnet.

[3] IP-Pakete die vor dem Enkapsulieren die maximale Paketgröße besitzen, würden dadurch fragmentiert werden. Von den meisten Anwendungen wird jedoch eine Erkennung der maximal übertragbaren Paketgröße (*MTU discovery*) durchgeführt [MD90], so daß die Größe der ausgesendeten Pakete angepaßt wird.

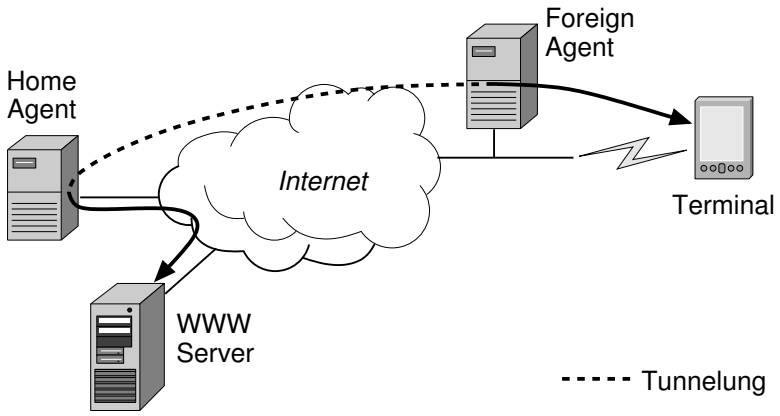


Abbildung 2.6: Mobile IP Routing mit IP-Tunnelung unter Verwendung eines Foreign Agent.

Diese IP-Adresse muß vom Home Agent aus erreichbar sein – im allgemeinen Fall darf dies also keine *private* IP-Adresse sein.

Seine temporäre Zustelladresse teilt der mobile Knoten seinem Home Agent mittels der Mobile IP-Registrierungsnachrichten mit. Dies muß jedesmal erfolgen, wenn sich die Zustelladresse ändert.

2.2.1 Handover und Netzwechsel

Die IETF definiert in [MK04] einen *Handover* als den Vorgang, bei dem ein mobiler Knoten seinen Netzzugangspunkt ändert oder zu ändern versucht. Nimmt man die dabei zurückgelegte (Netz-)topologische Entfernung zum Kriterium läßt sich ein Handover danach klassifizieren, ob der neue Zugangspunkt an die gleiche Routerschnittstelle angeschlossen ist (*Layer 2 Handover*), an eine andere Schnittstelle des gleichen Routers (*Intra Access Router Handover*), an einen anderen Router im gleichen (*Intra Access Network Handover*) oder einem anderen administrativen Bereich (*Inter Access Network Handover*).

Aus Sicht von Mobile IP versteht man unter einem *Netzwechsel* die durch den Wechsel des Netzzugangspunktes bedingte Änderung des IP-Netzes, die eine Anpassung des Routings für den mobilen Teilnehmer erfordert. In diesem Fall ist der mobile Knoten auch nicht mehr unter seiner bisherigen Zustelladresse erreichbar und muß daher seinem Home Agent umgehend seine neue Zustelladresse mitteilen.

In Abb. 2.7 ist der Nachrichtenfluß der Vermittlungsschicht im Falle eines Netzwechsels dargestellt. Nach dem Handover auf der Bitübertragungsschicht mit den entsprechenden spezifischen Prozeduren auf der Sicherungsschicht (siehe Abschnitt 3.1) erhält der mobile Knoten ein vom Foreign Agent periodisch ausgesendetes *Agent Advertisement*. Daraufhin sendet er dem Foreign Agent einen *Registration Request* für seinen Home Agent, um letzterem seine neue Erreichbarkeit anzuzeigen. Ist er selbst Tunnelendpunkt, sendet der mobile Knoten die ihm im aktuellen Netz zugewiesene IP-Adresse in einem *Registration Request* direkt an seinen Home Agent. Mit

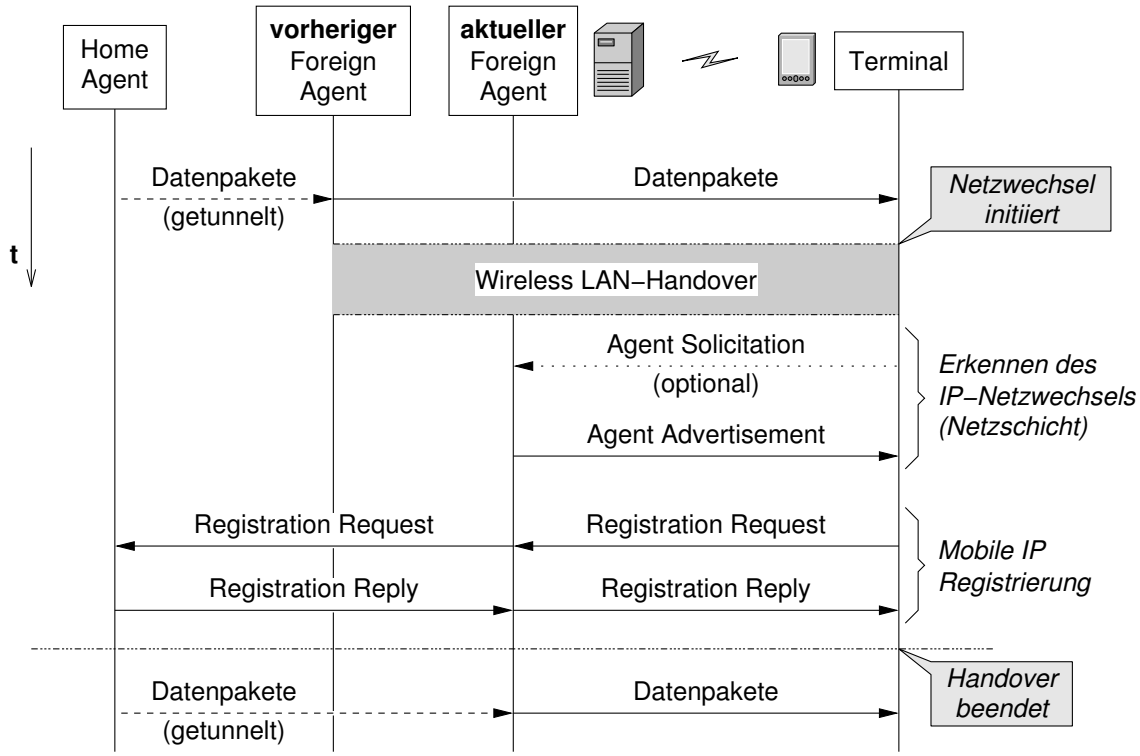


Abbildung 2.7: Nachrichtenflußdiagramm eines Netzwechsels mit Mobile IP-Registrierung.

einem *Registration Reply*, der wieder auf dem gleichen Weg zurück übertragen wird, bestätigt der Home Agent die Gültigkeit der Registrierung.

Im Falle eines *vertikalen* Handover kann typischerweise gleichzeitig mit dem bisherigen und dem zukünftigen Netz kommuniziert werden, so daß über die bisherige Anbindung Daten empfangen werden können während die Registrierung im neuen Netz durchgeführt wird. Dies ist bei einem *horizontalen* Netzwechsel nicht möglich, da eine Netzschnittstelle stets nur mit einem Netz kommunizieren kann. Ein Netzwechsel bedingt dann stets eine Unterbrechung der Datenkommunikation. Die Dauer eines Netzwechsels ist dabei die Zeitspanne, während der ein Teilnehmer nicht mehr unter seiner bisherigen IP-Adresse und noch nicht unter einer neuen IP-Adresse erreichbar ist.

Im Allgemeinen setzt sich die Dauer eines horizontalen Netzwechsels aus drei Komponenten zusammen, nämlich der Dauer des Wireless LAN Handover, der Latenz durch die Erkennung des neuen IP-Netzes und der Mobile IP-Registrierungsprozedur, wie in Abb. 2.7 dargestellt.

2.2.2 Erkennung von IP-Netzwechseln

Nicht jeder Handover zwischen zwei Basisstationen ist zugleich auch ein IP-Netzwechsel, da in einem IP-Netz mehrere Basisstationen vorhanden sein können. Um von der jeweils eingesetzten

Zugangstechnologie unabhängig zu sein, verläßt sich Mobile IP hier auf eigene Netzschichtnachrichten, die *Agent Advertisements*. Wann genau Mobile IP die Registrierung bei einem neuen Foreign Agent durchführt, wird dann anhand eines der drei folgenden Algorithmen [Per98] entschieden.

Lazy Cell Switching basiert auf der Gültigkeitsdauer der empfangenen *Agent Advertisements* (ICMP *Router Advertisements*). Das Terminal wertet alle eintreffenden *Router Advertisements* aus und speichert diese entsprechend der darin angegebenen Gültigkeitsdauer in einer Liste. Erst wenn zum aktuellen Foreign Agent kein gültiges *Advertisement* mehr eingetragen ist, wird ein anderer verfügbarer Foreign Agent kontaktiert. Dabei müssen jedoch nach dem letzten *Advertisements* des vorherigen Netzes mindestens drei aufeinanderfolgende *Agent Advertisements* des neuen Netzes empfangen werden. Dies ist für die Stabilität von Mobile IP wichtig, da möglichst verhindert werden muß, daß das mobile Terminal zwischen mehreren gleichzeitig erreichbaren Foreign Agents hin- und herschaltet. Diese können im gleichen Netz oder in unterschiedlichen, gleichzeitig erreichbaren Netzen sein.

Prefix Matching erlaubt den direkten Vergleich der (Sub-)Netzadressen zweier Foreign Agents. Unterscheidet sich die in den *Advertisements* angegebene Netzadresse von der des aktuell verwendeten Foreign Agent, so kann das Terminal auf einen Netzwechsel schließen. Auch hier müssen wiederum zwei weitere *Agent Advertisements* abgewartet werden, um unnötige Wechsel zwischen mehreren verfügbaren Foreign Agents mit unterschiedlichen Netzadressen zu vermeiden.

Eager Cell Switching führt ganz im Gegenteil sofort einen Handover zu einem neuen Foreign Agent durch, sobald ein *Agent Advertisement* von diesem empfangen wird. Dahinter steckt die optimistische Annahme, daß das *Advertisement* von einem Foreign Agent kommt, auf den sich das Terminal zubewegt. Dann wird auf diese Weise ein *make-before-break* Handover ermöglicht, da der Handover zu einem Zeitpunkt erfolgt zu dem noch Kontakt zum vorherigen Foreign Agent besteht. Dies kann jedoch nur dann funktionieren, wenn ein Teilnehmer von verschiedenen Basisstationen gleichzeitig Datagramme empfangen kann (was bei einigen Funktechnologien wie z. B. Wireless LAN nach IEEE 802.11 nicht der Fall ist).

Beschleunigen läßt sich die Erkennung des Netzwechsels, indem statt der zeitgesteuerten Erkennung[4] eine ereignisgesteuerte verwendet wird, wie es beispielsweise beim *Hinted Cell Switching* [FG01] vorgeschlagen wird. Dieser Ansatz setzt allerdings implizit voraus, daß es in jedem IP-Netz nur einen einzigen Foreign Agent gibt. Durch die Verwendung mehrerer Foreign Agents lassen sich jedoch Aspekte wie Lastteilung und eine erhöhte Ausfallsicherheit besonders einfach realisieren.

[4] Aufgrund der periodisch ausgesandten *Agent Advertisement*-Nachrichten haben die oben genannten Verfahren die Eigenschaften einer Zeitsteuerung.

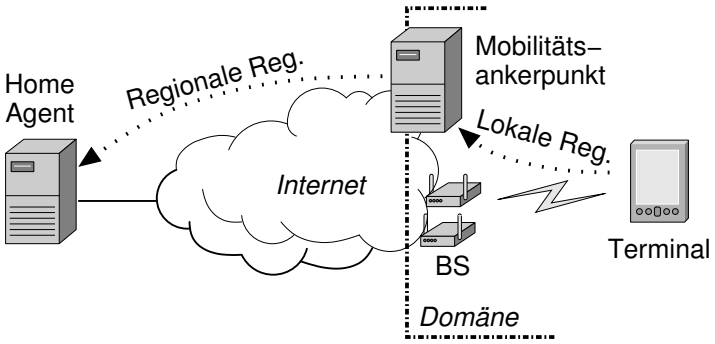


Abbildung 2.8: Lokale und regionale Registrierung beim Hierarchischen Mobile IP (IPv6).

2.2.3 Hierarchisches Mobile IP

Ein weiterer Aspekt beim Handover ist die Dauer der Mobile IP-Registrierungsprozedur. Für Mobile IPv4 wurde dazu in [FMM+99] das *Hierarchische Mobile IP* vorgeschlagen. Mittels einer Hierarchie (Baumstruktur) von Foreign Agents erübrigt sich dann eine Registrierung beim Home Agent, wenn der mobile Teilnehmer lediglich zwischen zwei Foreign Agents innerhalb einer Domäne wechselt. Die Registrierungsnachricht läuft bei diesem Verfahren nur bis zu demjenigen Foreign Agent, innerhalb dessen Zuständigkeitsbereichs sich das Terminal bewegt hat. Erst bei einem Wechsel in eine andere Domäne und somit in den Bereich eines anderen Wurzel-Foreign Agent muß der Home Agent wieder involviert werden. Auf diese Weise wird die Dauer der Mobile IP-Registrierungsprozedur um die Paketumlaufzeit zwischen dem schaltenden Foreign Agent – bei dem sich die Pfade der vorherigen und der neuen Registrierung kreuzen – und dem Home Agent verkürzt. Konzeptionell läßt sich dies als eine *Integration* von Mikro- und Makro-Mobilitätsunterstützung einordnen. Beim Hierarchischen Mobile IP muß jedoch der entsprechende Foreign Agent die Authentizität der Registrierungsnachrichten überprüfen, was eine deutliche Erweiterung seiner bislang auf Routingvorgänge beschränkten Funktionalität darstellt.

Ein dem Hierarchischen Mobile IPv4 ähnliches Verfahren wird in [SCMB05] für Mobile IPv6 vorgeschlagen. Hier ist jedoch lediglich eine zweistufige Hierarchie vorgesehen, bestehend aus dem *Mobilitätsankerpunkt* (MAP) und dem lokalen Zugangspunkt. Dabei übernimmt der MAP die Funktion eines lokalen Home Agent, indem sich die Teilnehmer bei diesem mit ihrer *lokalen* Zustelladresse (entsprechend dem lokalen Netzzugang) anmelden. Wie in Abb. 2.8 dargestellt generiert dieser wiederum eine Registrierungsnachricht an den Home Agent mit mit einer *regionalen* Zustelladresse. Kommt Routenoptimierung zum Einsatz (siehe Abschnitt 2.2.4), werden zusätzlich alle anderen Kommunikationspartner vom MAP benachrichtigt. Analog dem oben dargestellten Verfahren für Mobile IPv4 terminiert hier der MAP die Signalisierung bei Handovern innerhalb seiner Domäne.

Neben der Reduktion der Registrierungslatenz ist hier auch die Verringerung der Anzahl der versandten Nachrichten von Interesse, da z.B. bei Verwendung von Mobile IP mit Routenoptimierung jeder Kommunikationspartner über einen Ortswechsel (beziehungsweise die damit verbundene Änderung der IP-Adresse) informiert werden muß. Daß nicht jede Bewegung des Teil-

nehmers mitgeteilt werden muß ist dabei sicherlich auch im Sinne des Schutzes der Privatsphäre des Einzelnen.

2.2.4 Routenoptimierung für Mobile IP

Eines der Probleme beim regulären Mobile IP [Per02] ist die Tatsache, daß Datagramme zwischen dem mobilen Terminal und einer entfernten Gegenstelle stets über den Home Agent geleitet werden. Dies ist bedingt durch die Tatsache, daß die Gegenstelle nur die Heimatadresse des mobilen Teilnehmers kennt und auch nicht darüber informiert ist, wo sich dieser Teilnehmer aufhält. Die ungünstige Verkehrslenkung über den Home Agent resultiert in längeren Paketlaufzeiten sowie einer höheren Belastung der Transportnetze. Dies macht sich besonders in *Roaming*-Szenarien bemerkbar, also wenn der Teilnehmer sich weit entfernt von seinem Heimatnetz (und somit dem Home Agent) aufhält. Die Verwendung des Home Agent ist jedoch Voraussetzung für die Handoverunterstützung laufender Transportverbindungen.

Um diesen Umstand zu verbessern, wurde in [PJ98] die sogenannte *Routenoptimierung*[5] für Mobile IPv4 vorgeschlagen. Diese ermöglicht es, der Gegenstelle die aktuelle Zustelladresse des Teilnehmers mitzuteilen und so den Umweg über den Home Agent zu umgehen. Andererseits kann das gleiche Protokoll auch dafür verwandt werden, nach einem Handover dem vorherigen Foreign Agent die neue Zustelladresse mitzuteilen. Dadurch wird dieser in die Lage versetzt, Pakete an den neuen Aufenthaltsort weiterzuleiten und so die im Zuge des Handover auftretenden Paketverluste zu verringern.

Voraussetzung für die Routenoptimierung ist jedoch eine robuste *Authentifizierung* der Protokollnachrichten um zu verhindern, daß ein Angreifer die Datenströme eines anderen Nutzers umleitet. Dies ließe sich im Falle der Foreign Agents noch vergleichsweise einfach bewerkstelligen, da diese bereits in die Mobile IP-Signalisierung eingebunden sind. Die Involvierung der entfernten Gegenstellen in das Mobile IP-Protokoll erfordert jedoch neben der Modifikation der Protokoll-Logik (erforderlich ist eine Mobile IP-Implementierung) auch den Aufbau eines gemeinsamen Sicherheitskontextes, was einen Einsatz im größeren Maßstab verhindert.

Bei Mobile IPv6 [JPA04] stellt die Unterstützung der Routenoptimierung hingegen einen integralen Bestandteil des Protokolls dar. Dabei werden die Anforderungen an die entfernten Gegenstellen hinsichtlich der Authentifizierung der Protokollnachrichten durch eine Prüfung der Erreichbarkeit des mobilen Teilnehmers über beide Pfade (*Return Routability Check*) deutlich entschärft. Dazu sendet der mobile Teilnehmer, wie in Abb. 2.9 eingezeichnet, zwei unterschiedliche Nachrichten an die entfernte Gegenstelle, die eine über den Home Agent (*Home Test*) und die andere direkt unter Verwendung seiner Zustelladresse als Absenderadresse (*Care-of Test*). Die Gegenstelle (z. B. ein WWW-Server) wird nun beide Nachrichten kryptographisch verarbeiten und wieder an die jeweilige Absenderadresse zurücksenden. Nur dasjenige Terminal, welches beide Nachrichten empfängt, ist in der Lage daraus eine von der Gegenstelle akzeptierte Protokollnachricht zur Routenoptimierung (im Protokoll *Binding Update* genannt) zu erzeugen.

[5] Der Begriff “Optimierung” ist hier nicht im streng mathematischen Sinne zu verstehen.

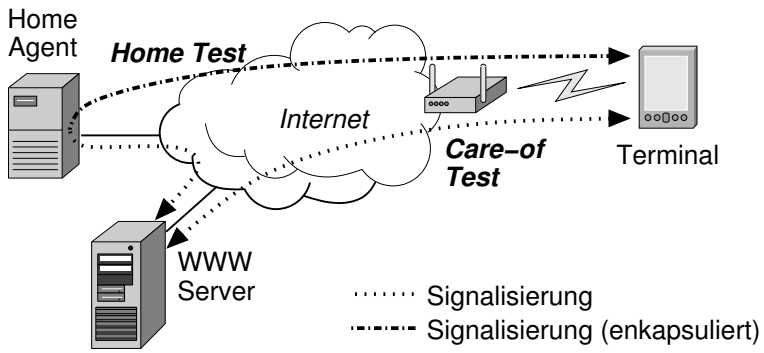


Abbildung 2.9: Signalisierungspfade für den *Return Routability Check*.

Diese Form der Authentifizierung kommt ohne globale Infrastruktur aus und gestattet somit die Routenoptimierung zwischen beliebigen Endgeräten. Wie in [NAA+05] dargestellt, entstehen durch dieses Verfahren keine zusätzlichen Sicherheitsprobleme gegenüber dem status quo. Jedoch setzt auch dieser Ansatz voraus, daß alle Endgeräte über eine entsprechende Protokollimplementierung verfügen. Indem Mobile IPv6 an die Einführung des IPv6 gekoppelt wird, hofft man diesem Problem begegnen zu können. Eine (theoretisch mögliche) Verbreitung dieses Verfahrens im bestehenden IPv4-basierten Internet erscheint daher unwahrscheinlich. Nachteilig bei dieser Form der Routenoptimierung ist jedoch, daß der mobile Teilnehmer im Falle eines Handover nicht nur an den Home Agent, sondern auch an alle Kommunikationspartner signalisieren muß. Dies erhöht den Signalisierungsaufwand beträchtlich.

2.2.5 Pre- und Post-Registrierung

Ein weiteres Problem beim Mobile IP-Handover besteht darin, daß die Erkennung auf der Netzschicht und die anschließende Registrierung beim Home Agent deutlich länger dauert als der Handover auf der Sicherungsschicht. Mittels der in [EM05] vorgeschlagenen Methoden zur Pre- und Post-Registrierung soll diesem begegnet werden.

Die *Pre-Registrierung* läßt sich nur dann durchführen, wenn das mobile Terminal oder der aktuelle Foreign Agent bereits vor dem Sicherungsschichthandover die Adresse des neuen Foreign Agent in Erfahrung bringen können. In diesem Falle kann das Terminal sich mittels einer entsprechenden Erweiterung des Mobile IP Registrierungsprotokolls bereits vor dem Kanalwechsel beim zu erwartenden neuen Foreign Agent registrieren. Im Idealfall läßt sich auf diese Weise die durch Mobile IP verursachte Latenz beim Handover vollständig vermeiden.

Die *Post-Registrierung* wiederum richtet nach einem Sicherungsschichthandover des mobilen Terminals einen Tunnel zwischen dem alten und dem neuen Foreign Agent (bei dem sich das Terminal aber noch nicht registriert hat) ein. Dies geschieht wiederum auf der Grundlage von Benachrichtigungen der Sicherungsschicht, die dem neuen oder dem alten Foreign Agent die Adresse des jeweils Anderen mitteilen. Der Foreign Agent im vorherigen Netz wird dann alle Pakete für den Teilnehmer durch den Tunnel weiterleiten, so daß der Paketverlust durch den

Handover verringert wird. Während er weiterhin Pakete senden und empfangen kann, wird der Teilnehmer sich beim neuen Foreign Agent registrieren. Da der Tunnelaufbau ohne Zutun des Terminals stattfindet, ist eine gegenseitige Authentifizierung der Foreign Agents unumgänglich.

Für Mobile IPv6 wurde in [Koo05] ein weitgehend analoges Verfahren vorgeschlagen. Entsprechend kann der Teilnehmer vor dem bevorstehenden Handover bereits eine Zustelladresse im zukünftigen Netz beziehen. Nach dem Handover kann er beim Zugangsrouter, dessen Adresse er bereits vorher festgestellt hat, die schnelle Tunnelung der Pakete an den neuen Aufenthaltsort einleiten. Wie auch oben hängt diese Funktionalität davon ab, ob die Sicherungsschicht der verwendeten Netzzugangstechnologie Informationen über den nächsten zu erwartenden Netzzugangspunkt und dann auch den eigentlichen Handover übergibt.

Auf die Anforderungen an die Schnittstellen zur Sicherungsschicht wird in [McC05] näher eingegangen. Um die oben beschriebenen Methoden ausführen zu können, muß der Mobile IP-Client im Terminal nicht nur eine Liste der verfügbaren Zugangsrouter und deren aktueller Signalstärke bekommen können. Er muß auch in der Lage sein den Handover zu einem bestimmten Netz steuern zu können, auch wenn dieses nicht mehr das stärkste empfangbare Netz sein sollte. Daneben ist eine sofortige Benachrichtigung der Netzschicht nach Fertigstellung des Sicherungsschichthandover von Nöten, um eine unverzügliche Wiederaufnahme der Paketübertragung zu gewährleisten.

2.2.6 Proxy Mobile IP

Um Mobilitätsunterstützung auch für Terminals anzubieten, die nicht über eine Implementierung von Mobile IP verfügen, wurde von Cisco das *Proxy Mobile IP* [LDY06] bei der IETF eingebracht. Hierbei erledigt ein *Mobilitäts-Proxy-Agent* (MPA) die Registrierung beim Home Agent für den mobilen Teilnehmer. Hierbei gibt es drei Szenarien:

- Netzseitiger MPA in der Basisstation (BS)
- Netzseitiger MPA im Zugangsrouter oder Foreign Agent
- Mobilseitiger MPA in Form eines Terminierungs- oder Adaptierungselementes

Wie in Abb. 2.10 für den Fall des netzseitigen, mit dem Foreign Agent (FA) kollokierten MPA dargestellt, registriert sich der von seiner Mobilität nichts ahnende Teilnehmer regulär bei einer Basisstation. Diese Registrierung wird durch entsprechende Mechanismen, z. B. einen AAA-Server, verifiziert. Die Basisstation wird daraufhin die Zugangsdaten des Teilnehmers an den MPA/FA übermitteln, der wiederum die Registrierung beim Home Agent einleitet. Datenpakete für den mobilen Teilnehmer werden vom MPA/FA dekapsuliert und an das Terminal übermittelt. Vom Terminal ausgesandte Pakete wird der MPA/FA mittels *Proxy ARP* empfangen, wobei er sich hier aus Sicht des Terminals wie das vom Terminal adressierte Standardgateway verhält.

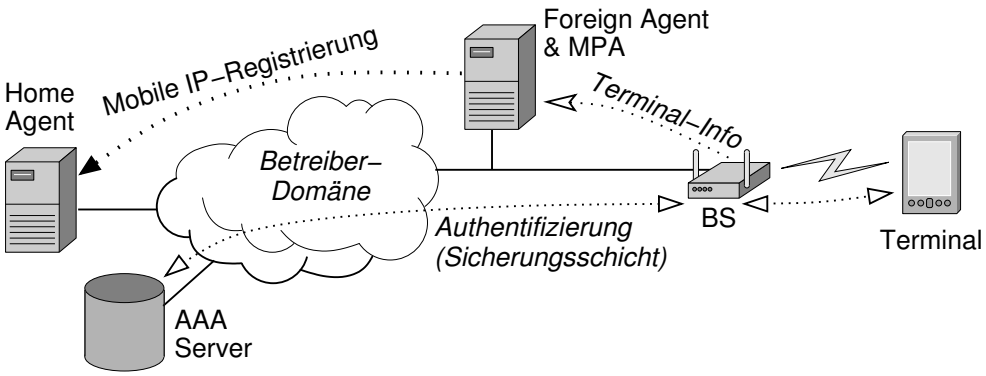


Abbildung 2.10: Sicherungsschicht- und Mobile IP-Registrierung beim *Proxy Mobile IP*.

Indem das Terminal bei netzseitigem MPA nicht die Mobile IP-Signalisierung ausführen muß, wird der drahtlose Zugang von diesem Datenaufkommen entlastet. Voraussetzung dafür ist jedoch, daß der MPA das vollkommene Vertrauen des Home Agent hinsichtlich der Datenströme des Terminals genießt. Dies schränkt die Verwendung der beiden netzseitigen Varianten des *Proxy Mobile IP* ein auf die Domäne eines einzigen Betreibers oder einiger weniger Betreiber mit wechselseitigen Abkommen.

2.3 Zwischenschicht zur Mobilitätsunterstützung

Bei Mobile IP ist der Home Agent der Dreh- und Angelpunkt der Kommunikation, der die Datagramme an den aktuellen Aufenthaltsort weiterleitet. Dadurch wird dem Umstand begegnet, daß eine IP-Adresse einerseits ein Terminal eindeutig identifiziert und andererseits zugleich auch seinen Ort angibt. Die beiden im folgenden vorgestellten Lösungsansätze zur Mobilitätsunterstützung auf der Netzschicht entkoppeln die Transportschicht von der ortsangebenden IP-Adresse, indem eine zusätzliche Abstraktionsebene eingeführt wird.

2.3.1 Mobilität mit dem Host Identity Protocol (HIP)

Das in [MNJH06] vorgeschlagene *Host Identity Protocol* (HIP) vergibt den Teilnehmern zusätzlich zu Ihrer wechselnden IP-Adresse dauerhafte Identitäten. Transportverbindungen binden sich an eine der Identitäten des Terminals und überlassen es dem HIP, die zugehörige IP-Adresse zu ermitteln. Ändert sich die IP-Adresse einer der beiden Gegenstellen im Laufe der Kommunikation, so wird die Zuordnung von Identität und IP-Adresse dynamisch geändert [MN06, Hen06].

Im Vergleich zu den vom *Domain Name System* (DNS) verwalteten Rechnernamen ist die Zuordnung von IP-Adresse und Identität(en) weitaus flexibler und auch schnell genug zur Unterstützung von Mobilität. Andererseits könnten bestehenden Transportverbindungen allein auf Basis eines dynamischen Rechnernamens nicht weitergeführt werden, da dieser beim Verbindungsaufbau in eine IP-Adresse umgewandelt wird an die sich die Transportverbindung dann fest bindet.

Gleich ist beiden Systemen der Bedarf für eine Infrastruktur. Insbesondere für den Fall, daß beide Gegenstellen zugleich einen Handover durchführen, sind sogenannte *Rendezvous-Punkte* unerläßlich, um die neue IP-Adresse des jeweils Anderen in Erfahrung zu bringen.

2.3.2 Internet Indirection Infrastructure (i3)

Basis der in [SAZ+04] vorgeschlagenen *Internet Indirection Infrastructure* (*i3*) ist ein Overlaynetz, welches von einem Satz verteilter Server aufgespannt wird. Teilnehmer registrieren sich von ihrem neuen Aufenthaltsort aus und können dann an ihre Kennung gesendete Pakete empfangen. Die *i3* orientiert sich an Peer-to-Peer-Netzen, deren besonderes Kennzeichen die verteilten Suchverfahren für andere Objekte (Teilnehmer, Dateien, etc.) sind. Aus dem Blickwinkel der Mobile IP-Welt stellen die *i3*-Server einen selbstorganisierend verteilten Home Agent dar.

Wie auch das HIP setzt die *i3* auf der vorhandenen IP-Infrastruktur (ergänzt um einige zusätzliche Einheiten) auf. Beide erfordern jedoch Änderungen nicht nur bei jedem mobilen Teilnehmer, sondern auch bei allen potentiellen Kommunikationspartnern. Einerseits muß eine entsprechende Protokollimplementierung vorhanden sein, andererseits müssen aber auch die Transportschichtprotokolle und unter Umständen auch einige Anwendungsprogramme angepaßt werden. Dies sind faktisch unüberwindliche Hindernisse, die einer Einführung im heutigen IPv4-basierten Internet entgegenstehen.

2.4 Mobilitätsunterstützung auf der Transportschicht

Ein inhärenter Vorteil von Transportschichtansätzen zur Mobilitätsunterstützung ist, daß die Flußkontrolle leicht mit eingebunden werden kann und insgesamt eine größere Nähe zum Anwendungsprogramm besteht. Die strikte Trennung der Schichten ist einer der Vorteile und zugleich eines der Probleme von Mobile IP. Beispielsweise verringert die Flußkontrolle der Transportschicht (z. B. von TCP) die Datenrate nach einem Mobile IP-Handover, da fehlende Pakete als Symptom für eine Netzüberlastung fehlinterpretiert werden[6].

Auf der Transportschicht gibt es die beiden etablierten Protokolle TCP und UDP sowie das vergleichsweise neue SCTP. Die Mobilitätsunterstützung auf der Transportschicht ist dabei für alle Protokolle und Anwendungen transparent, die im Sinne der Subsidiarität der Schichten konzipiert sind, also insbesondere selbst keine IP-Adressen verwenden.

2.4.1 TCP-Mobilität

Bei dem in [SB00] vorgeschlagenen Ansatz werden neue TCP-Optionen eingeführt, die die Migration laufender Verbindungen nach einem Handover gestatten. Die Erreichbarkeit des Teilnehmers für neue eingehende Verbindungen wird durch dynamisches DNS sichergestellt. Dabei

[6] Der durch Bitfehler des Funkkanals verursachte Paketverlust wird z. B. bei Wireless LAN (IEEE 802.11) bereits von der Sicherungsschicht durch Sendewiederholungen behoben.

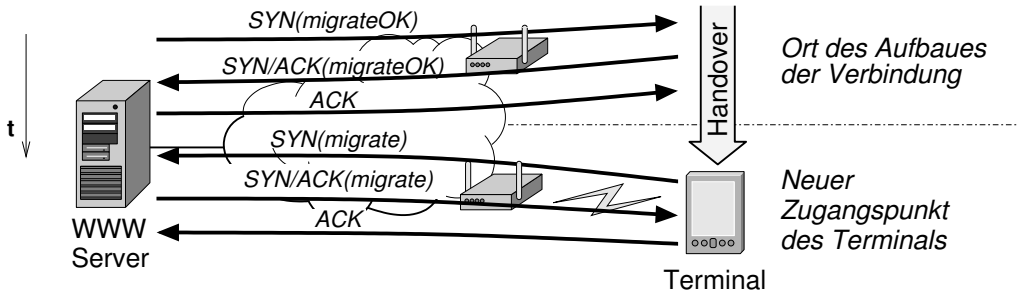


Abbildung 2.11: Aufbau und Migration einer TCP-Verbindung nach [SB00].

wird der DNS-Eintrag nach jeder Änderung der IP-Adresse des Teilnehmers stets umgehend über ein authentifiziertes Protokoll aktualisiert. Die DNS-Antworten des Servers an andere Terminals werden überdies mit einer sehr kurzen Verfallsdauer versehen, um eine rasche Aktualisierung nach einem Handover des mobilen Teilnehmers zu gewährleisten.

Beim Verbindungsaufbau wird mittels zusätzlicher – zum herkömmlichen TCP kompatibler – Optionen festgestellt, ob beide Kommunikationspartner die neuartigen Funktionen zur Migration unterstützen. Ist dies der Fall, wird zusätzlich eine 64-Bit-Marke (*Token*) ausgetauscht, die nach einem Handover zur eindeutigen Identifikation der Verbindung benötigt wird. Der mobile Teilnehmer sendet dann eine neue Synchronisationsnachricht (TCP SYN) unter Verwendung seiner neuen IP-Adresse und der Identifikationsmarke, wie in Abb. 2.11 dargestellt. Nachdem die Gegenstelle deren Authentizität geprüft hat, wird sie die TCP-Segmente nun an die neue Adresse senden, wobei alle weiteren Eigenschaften der TCP-Verbindung (insbesondere die Zustände der Flußkontrolle) beibehalten werden. Von Nachteil ist jedoch, daß zur Aktualisierung der Adresse die Dreiwege-Verbindungsaufbauprozedur des TCP wiederholt wird, während der keinerlei Daten übertragen werden können.

2.4.2 Mobile SCTP

Das *Stream Control Transport Protocol* (SCTP) [OY02] ist eine Weiterentwicklung des TCP mit der bemerkenswerten Eigenschaft, daß es sich an mehrere Netzschnittstellen eines Terminals binden läßt. Fällt eine Schnittstelle (oder der Pfad dorthin) aus, kann das SCTP stattdessen eine der anderen Schnittstellen verwenden, ohne daß die Anwendung davon berührt würde. Daneben sind mehrere parallele Unterdatenströme möglich, deren Übertragung von derselben Flußkontrolle gesteuert wird. Dadurch lassen sich die beim TCP auftretenden Blockierungseffekte vermeiden. Erweiterungen [SRX+06] des SCTP gestatten es, die mit einer Verbindung assoziierten IP-Adressen zu ändern.

In [XKWM02, RT06] wird gezeigt, wie sich damit *make-before-break* Handover unterstützen lassen. Dazu wird vor einem Handover die im neuen Netz zugewiesene IP-Adresse über die noch bestehende Verbindung zum alten Netz signalisiert und so der Gegenstelle mitgeteilt, ohne daß hierzu besondere Sicherheitsmechanismen nötig wären. Dieser Ansatz versagt jedoch bei

einem unvorhergesehenen *break-before-make* Handover. In diesem Falle kann die Aktualisierung über eine bestehende Mobile IP-Infrastruktur ausgeführt werden [FA04]. Dazu wird beim Verbindungsaufbau nicht nur die temporäre IP-Adresse des besuchten Netzes, sondern auch die dauerhafte Heimat-IP-Adresse bei der Gegenstelle registriert. Eine zusätzliche Logik lenkt die Datenströme dann bevorzugt über die temporäre IP-Adresse, um den Home Agent zu entlasten und unnötige Paketlaufzeiten zu vermeiden.

2.4.3 MSOCKS

Im Gegensatz zu den beiden soeben dargestellten Methoden zur Mobilitätsunterstützung auf der Transportschicht wird bei MSOCKS [MB98] die Mobilität des Teilnehmers vor der Gegenstelle vollständig verborgen. Dazu baut das mobile Terminal seine Transportverbindung von einem Proxy aus auf, der als fester Zwischenpunkt für die Kommunikation mit anderen Teilnehmern dient. Auf diese Weise erfolgt der Datentransport zwischen mobilem Terminal und fester Gegenstelle über zwei hintereinanderliegende TCP-Verbindungen. Um die Ende-zu-Ende-Semantik der Transportbeziehung zu wahren, werden die beiden TCP-Verbindungen miteinander "verspleißt". Dies hat zur Folge, daß Bestätigungen für Segmente erst dann versandt werden, nachdem die jeweilige Gegenstelle den Empfang bestätigt hat.

Nach einem Handover baut das mobile Terminal eine neue Transportverbindung zum Proxy auf. Kann es sich erfolgreich authentifizieren, ermöglicht ihm der Proxy, diese neue Transportverbindung mit der noch bestehenden zu verspleißen. Auf diese Weise kann die Kommunikation für den festen Kommunikationspartner vollkommen transparent wiederaufgenommen werden. Segmente, die das mobile Terminal zuvor noch nicht bestätigt hat, werden dann von der Gegenstelle erneut versandt.

2.5 Mobilitätsunterstützung in höheren Schichten

Siedelt man die Mobilitätsunterstützung oberhalb der Transportschicht an, läßt sich diese besser an die Erfordernisse der Anwendung abstimmen. Allerdings macht dies zugleich eine separate Mobilitätsunterstützung für jede Anwendung erforderlich. Um die Performanz der Verfahren auf niederen Schichten zu erreichen, ist eine Kommunikation mit ebendiesen niederen Schichten erforderlich.

2.5.1 Mobile SIP

Beim *Session Initiation Protocol* (SIP) [RSC+02] registriert sich ein Teilnehmer bei seinem zuständigen SIP-Proxy mit der aktuellen IP-Adresse, um für eingehende Verbindungen erreichbar zu sein. Im Zuge des Verbindungsaufbaues leitet der SIP-Proxy der rufenden Seite die INVITE-Nachricht an den für den gerufenen Teilnehmer zuständigen SIP-Proxy weiter, der diese schließlich jenem Teilnehmer zustellt. Wie in Abb. 2.12 gezeigt, antwortet das gerufene Endgerät mit

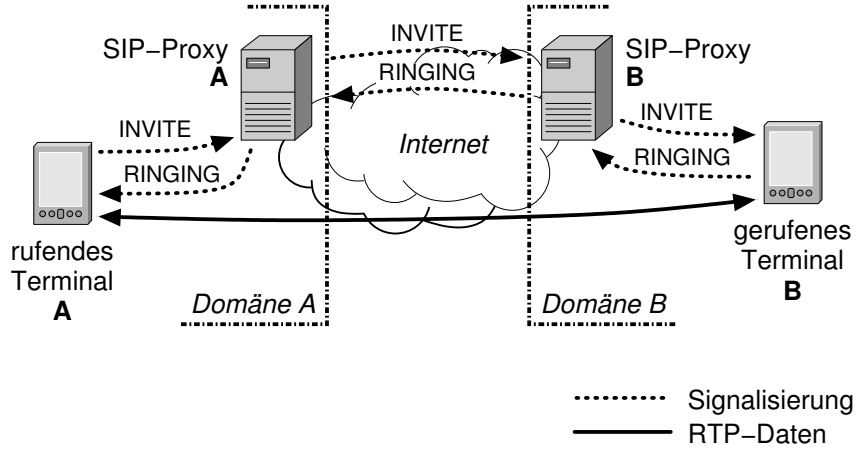


Abbildung 2.12: Vereinfachter SIP Verbindungsaufbau unter Verwendung von SIP-Proxys.

einer RINGING-Antwort. Diese beinhaltet die eigene IP-Adresse und ermöglicht daher, daß die Nutzdaten (z. B. RTP-Pakete mit den Sprachdaten) auf dem direkten Routingpfad – also nicht notwendigerweise über den SIP-Proxy – transportiert werden, ähnlich wie beim Mobile IP mit Routenoptimierung die Daten nicht zwingend über den Home Agent geleitet werden. Da beispielsweise bei der Mobiltelephonie die Zustände eines Rufes ohnehin von Netzseite her überwacht werden müssen, verwendet das *Roaming* beim *IP Multimedia Subsystem* (IMS) [3GP06] des UMTS diese Lösung.

Wechselt der Teilnehmer während einer Verbindung in ein anderes Netz, gestattet ihm *Mobile SIP* [SW00] dies seinem Kommunikationspartner mittels einer INVITE-Nachricht mitzuteilen, die selbstverständlich entsprechend authentifiziert zu sein hat. Dieser wird daraufhin seine Datenpakete an die neue Adresse senden. Die minimale Unterbrechung der Kommunikation entspricht der Paketumlaufzeit zwischen den beiden Terminals und läßt sich durch die Verwendung von dazwischenliegenden RTP-Proxys verkürzen.

Im Falle von *make-before-break* Handovern, insbesondere vertikalen Handovern, läßt sich der ebenfalls in [SW00] vorgestellte Mechanismus zur Sitzungsmobilität nutzen. Dabei verweist der Teilnehmer mittels einer REFER-Nachricht die Gegenstelle an eine andere SIP-Adresse, die zu einer anderen Schnittstelle des gleichen (oder auch eines anderen) Gerätes gehört[7]. Daraufhin wird die Gegenstelle mit dieser neuen Adresse eine gleichwertige Sitzung aushandeln und auch soweit möglich aufbauen. Der inhärente Vorteil dieses Verfahrens bei heterogenen Netzen ist, daß hierbei die Sitzungsparameter für das neue Transportnetz erneut ausgehandelt werden.

[7]Beispielsweise kann einem Terminal eine SIP-Kennung für das UMTS-Netz und eine andere für den Wireless LAN-Zugang zugeordnet sein.

2.5.2 Resilient Mobile Socket (RMS)

Das Mobile SIP ermöglicht es einem Terminal, eine Sitzung an den neuen Aufenthaltsort nachzuführen oder auch über ein zweites Netz parallel eine weitere Sitzung zu eröffnen. Wie nun diese Möglichkeiten der Signalisierung möglichst wirksam genutzt werden, um Dienste für mobile Teilnehmer zu erbringen, bleibt dabei dem Anwendungsprogramm überlassen.

In [KP06] wird mit dem *Resilient Mobile Socket* (RMS) ein Verfahren eingeführt, um Mobile SIP für herkömmliche, mobilitätsagnostische Anwendungsprogramme nutzen zu können. Ein Mobilitätsmanager sowie ein ihm zugeordneter Netzschnittstellenmanager wählen dabei den jeweils besten Netzzugang aus. Um die dadurch bedingte Änderung der IP-Adressen in den Datenpaketen vor der Anwendung zu verbergen, werden die für die betreffende Anwendung bestimmten Datagramme – ähnlich einer NAT – modifiziert.

2.5.3 Mobile People Architecture

Zusätzlich zu den bisherigen ISO/OSI-Schichten wird in [MRS+99] die sogenannte Personenschicht eingeführt. Dadurch spiegelt sich die Tatsache explizit im Schichtenmodell wieder, daß die Kommunikation zwischen zwei Personen nicht nur über verschiedene Transportmedien, sondern auch über unterschiedliche Anwendungen ausgeführt werden kann. Die Adressierung erfolgt hier anhand einer *Personal Online ID*, die jede Person eindeutig identifiziert.

Ein *Persönlicher Proxy* organisiert die Kommunikation über verschiedene Anwendungen hinweg. Dabei erfüllt er vier Kernaufgaben: Nachverfolgen des Teilnehmers (*Tracking*), Umsetzen von Zustellregeln (*Rules Engine*), Zustellung (*Dispatcher*) und anwendungsspezifische Bearbeitung (*Application Drivers*). Dabei paßt der *Persönliche Proxy* die Kommunikation an die Bedürfnisse der betreffenden Person hinsichtlich Art und Umstand der Kommunikation an, wobei entsprechend den Nutzerpräferenzen beispielsweise auch eingehende Rufe abgewiesen oder auf eine andere Anwendung (z. B. Anrufbeantworter) umgeleitet werden können.

2.6 Mikro-Mobilitätsunterstützung

Unter dem Begriff *Mikro-Mobilität* versteht man allgemein die Mobilität eines Terminals innerhalb eines beschränkten Bereiches [MK04]. Dieser kann ein einziges IP-Netz oder eine Domäne von IP-Netzen umfassen. Protokolle zur Makro-Mobilitätsunterstützung bringen im Falle kleiner, aber häufiger Ortswechsel dreierlei Probleme mit sich [KLR+06a]: Lange Aktualisierungsdauer, hoher Signalisierungsaufwand und mangelnder Schutz der Privatsphäre, da beispielsweise im Falle von Mobile IP jeder Netzwechsel dem Home Agent mitgeteilt wird.

Charakterisierend für alle Verfahren zur Mikro-Mobilitätsunterstützung ist, daß diese den Bereich einer Domäne bis zu einem dedizierten Grenzknoten umfassen und jegliche Bewegung

innerhalb der Domäne höchstens bis zu diesem signalisiert wird. Dadurch verkürzt sich automatisch die Aktualisierungsdauer im Falle eines Handover, auch der Schutz der Privatsphäre wird erhöht.

Im folgenden werden nun exemplarisch drei Verfahren zur Mikro-Mobilitätsunterstützung vorgestellt: *Cellular IP* [Val99] arbeitet auf der Sicherungsschicht, auf der Netzschicht wiederum wirken die Mechanismen von *Hawaii* [RVS+02] auf Basis von Unicast und *Mombasa* [Fes03], welches im Gegensatz dazu Multicast verwendet.

2.6.1 Cellular IP

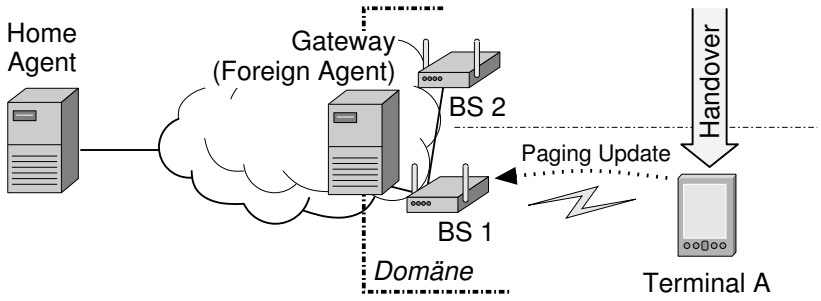
Um die Skalierbarkeit für große Nutzerzahlen zu ermöglichen, wird bei *Cellular IP* [Val99] zwischen aktiven und inaktiven Terminals unterschieden, wobei angenommen wird, daß nur ein kleiner Teil der Terminals zeitgleich aktiv ist. Aktive Terminals sind solche, die Datagramme senden oder empfangen. Inaktive Terminals hingegen senden oder empfangen keine Nutzdaten, sollen aber dennoch für eingehende Verbindungen erreichbar sein.

Cellular IP ist vollkommen transparent für Protokolle zur Makro-Mobilitätsunterstützung, wie beispielsweise Mobile IP. Ein Gateway fungiert dabei als Foreign Agent und terminiert die ankommenden IP-Tunnel. Innerhalb der *Cellular IP*-Domäne werden dann die Heimat-IP-Adressen der Terminals zu deren Unterscheidung verwandt. Auf dem mobilen Terminal ist lediglich eine zusätzliche Signalisierungsinstanz erforderlich. Insbesondere benötigt *Cellular IP* keine weitergehende Konfiguration oder zusätzliche Authentifizierung, was für eine Mikro-Mobilitätsunterstützung gemeinhin als erstrebenswert angesehen wird [KLR+06b].

Datagramme werden innerhalb der *Cellular IP*-Domäne über ein Baumnetz mittels eines flachen Routings zu den Teilnehmern gelenkt. Dazu dienen an den Verzweigungspunkten eingerichtete *Routing Caches*, die eine Liste aller an dem entsprechenden Ast angeschlossener aktiver Terminals beinhalten. Dabei kann ein Teilnehmer im Falle eines *make-before-break* Handover auch an zwei Ästen zugleich angemeldet sein. Daher haben die Einträge in den verteilten *Routing Caches* eine vergleichsweise kurze Lebensdauer in der Größenordnung einiger Paketzwischenankunftszeiten.

Um den Signalisierungsaufwand für inaktive Teilnehmer so gering wie möglich zu halten, existieren parallel dazu die *Paging Caches*. Deren Einträge haben eine deutlich längere Lebensdauer, so daß die Zahl der Signalisierungsnachrichten sowohl für unbewegte Teilnehmer als auch für mobile Teilnehmer verringert wird, die mehrfach zwischen bestimmten Zellen wechseln. Wie in Abb. 2.13 dargestellt sendet das inaktive Terminal nach einem Handover in den Bereich einer anderen Basisstation eine *Paging Update*-Nachricht, die dem *Paging Cache* hinzugefügt wird. In Abb. 2.13 sind die *Paging Cache*-Einträge unmittelbar nach einem Handover eingetragen. Da es sich um ein inaktives Terminal handelt, sind in den *Routing Caches* keine Einträge für dieses Terminal vorhanden.

Trifft nun ein Paket für einen Teilnehmer ein, zu dem es keinerlei *Routing Cache*-Einträge gibt, so werden *Paging*-Nachrichten über diejenigen Äste versandt, auf die entsprechende *Paging Cache*-Einträge zeigen. Meldet sich der Teilnehmer dann mit einer entsprechenden Antwort (*Route*



Paging Cache Gateway	
Terminal A	BS 1

Paging Cache BS 1	
Terminal A	BS 2, lokal

Paging Cache BS 2	
Terminal A	lokal

Abbildung 2.13: Handover eines inaktiven Terminals beim *Cellular IP* mit *Cache*-Einträgen.

Update), werden zu seinem aktuellen Aufenthaltsort *Routing Cache*-Einträge erzeugt, so daß ihm ankommende Datagramme zugestellt werden können.

2.6.2 Handoff-Aware Wireless Access Internet Infrastructure (Hawaii)

Im Gegensatz zum oben beschriebenen *Cellular IP* gründet die *Handoff-Aware Wireless Access Internet Infrastructure* (Hawaii) [RVS+02] auf einem Netzschichtansatz. Diese unterstützt die Mobilität eines Terminals innerhalb einer aus mehreren IP-Netzen bestehenden Zugangsdomäne. Da dabei ein zum Zwecke der Mobilitätsunterstützung modifiziertes Routing zum Einsatz kommt, kann auch ein vermaschtes Netz genutzt werden.

Meldet sich ein neuer Teilnehmer in der Zugangsdomäne an, so erhält er eine lokale IP-Adresse aus dem entsprechenden IP-Netz zugewiesen. Mit dieser kann er sich auch bei seinem Home Agent anmelden. Wechselt das Terminal nun in ein anderes IP-Netz der gleichen Domäne, sendet es eine entsprechende Aktualisierungsnachricht an den für dieses Netz zuständigen Router. Dieser informiert dann weitere Router über den neuen Aufenthaltsort des Terminals. Diese werden dann mittels spezifischer Routingeinträge dafür Sorge tragen, daß wieder alle Datagramme den mobilen Teilnehmer erreichen.

Verglichen mit Mobile IP ist der Aufwand für einen Handover geringer, sowohl hinsichtlich der Zahl der Signalisierungsnachrichten als auch des Berechnungsaufwandes. Im Gegensatz zu *Cellular IP* ist dieser Ansatz offen für eine Reservierung von Ressourcen, beispielsweise mittels RSVP [BZB+97], auch entfällt der *Paging*-Vorgang.

2.6.3 Mobility Support – A Multicast-Based Approach (Mombasa)

Der IP-Multicast stellt bereits eine Infrastruktur zur ortsunabhängigen Adressierung zur Verfügung. Beim *Mobility Support – A Multicast-Based Approach* (Mombasa) [FWW02, Fes03] wird dies dahingehend zur Unterstützung von Mikro-Mobilität genutzt, daß jedem Terminal innerhalb

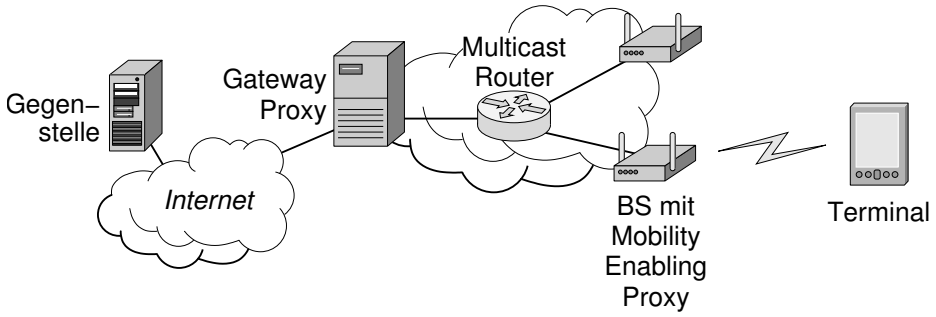


Abbildung 2.14: *Mombasa* [FWW02] zur Multicast-basierten Mikro-Mobilitätsunterstützung.

einer Domäne von IP-Netzen mit Multicast-fähigen Routern eine bestimmte Multicast-Adresse zugeordnet wird. Wie in Abb. 2.14 gezeigt erreichen eingehende Datagramme zunächst den *Gateway Proxy*, durchlaufen dann das Multicast-Routing und werden schließlich vom *Mobility-Enabling Proxy* dem mobilen Terminal zugestellt.

Eingehende Datagramme an die Unicast-Adresse eines Terminals werden vom *Gateway Proxy* auf dessen Multicast-Adresse umadressiert. Das Terminal subskribiert diese Multicast-Gruppe an seinem jeweiligen Aufenthaltsort beim jeweils zuständigen *Mobility-Enabling Proxy*, der zugleich die Unicast-Adresse des Terminals in den IP-Paketen wiederherstellt. Das Mobilitätsmanagement wird so vollständig von den dazwischenliegenden Multicast-Routern übernommen.

Die Eigenschaft, daß Multicast auch Datagramme an mehrere Empfänger zustellen kann, kann im Falle von Handovern genutzt werden. Im einfachen Fall eines *make-before-break* Handover meldet sich das Terminal zunächst im neuen *Mobility-Enabling Proxy* an und bricht die Verbindung mit dem vorherigen Netzzugang erst ab, nachdem es die ersten Pakete aus dem neuen Netz empfangen hat. Auf diese Weise sind verlustlose Handover möglich.

Daneben lassen sich prädiktive Handover realisieren. Hierbei wird eine indirekte Registrierung des Teilnehmers bei benachbarten *Mobility-Enabling Proxys* eingeleitet. Diese erhalten daraufhin zwar alle für den Teilnehmer bestimmten Datagramme, speichern diese jedoch in einem Ringpuffer zwischen. Erst wenn der Teilnehmer sich bei einem dieser *Mobility-Enabling Proxys* anmeldet, sendet dieser die dann gerade gepufferten IP-Pakete aus. Ist die Größe des Ringpuffers passend gewählt, so sind auch auf diese Weise verlustfreie Handover möglich. Die dafür erforderlichen Sicherheitsmaßnahme werden in [WRSW04] dargelegt und basieren auf Verwendung eines lokalen AAA-Servers.

2.7 Vergleich und Bewertung

2.7.1 Verfahren zur Makro-Mobilitätsunterstützung und deren Einsetzbarkeit

Eine praktikable Lösung zur Makro-Mobilitätsunterstützung soll in heutigen Netzen mit dem geringstmöglichen Aufwand sofort einsetzbar und für alle relevanten Anwendungen nutzbar sein. Daraus lassen sich folgende Kriterien für die Beurteilung der verschiedenen in diesem Kapitel vorgestellten Ansätze ableiten:

- Geringstmögliche Änderungen in den Netzen, wobei Änderungen in den Transportnetzen (das heißt außerhalb der Zugangsnetze) besonders aufwendig sind,
- Kompatibilität mit den vorhandenen Gegenstellen (z. B. öffentliche WWW-Server),
- Eingehende Verbindungen müssen für bestimmte Anwendungen (z. B. *Voice over IP*) möglich sein[8],
- Kompatibilität mit Mikro-Mobilitätsverfahren, da sich Anwendungen bei zu häufigen und zu langen Unterbrechungen typischerweise nicht mehr nutzen lassen.

In Tab. 2.2 werden die verschiedenen Ansätze zur Makro-Mobilitätsunterstützung hinsichtlich dieser Kriterien miteinander verglichen. Modifikationen am mobilen Terminal selbst sind als unproblematisch einzustufen, da im Gegensatz zu öffentlichen Servern der Eigentümer selbst die Mobilitätsunterstützung wünscht.

Am problematischsten hinsichtlich der genannten Kriterien ist es sicherlich, eine neue Schicht zwischen Netz- und Transportschicht einzufügen (hier als Schicht $3\frac{1}{2}$ bezeichnet). Sowohl das *Host Identity Protocol* (HIP) als auch die *Internet Indirection Infrastructure* (*i3*) setzen voraus, daß die Implementierung des neuen Protokolls sowohl bei allen mobilen Teilnehmern als auch allen potentiellen Kommunikationspartnern verfügbar ist. Damit verbunden ist eine Änderung der TCP/UDP-Implementierungen und auch derjenigen Anwendungen, die entgegen den allgemeinen Empfehlungen – wie beispielsweise [Sen02] – direkt IP-Adressen einsetzen. Auf Netzseite verlangen beide Ansätze unterstützende Server, wobei deren Ort und Anzahl den Anforderungen an die Performanz entsprechend gewählt werden kann.

Im Gegensatz dazu benötigen die Endgeräte-basierten Ansätze – TCP-Mobilität und Mobile SCTP – nur geringe Unterstützung durch das Netz. Das Terminal bleibt durch dynamisches DNS [VTRB97] erreichbar, jedoch bricht die Verbindung ab wenn sich beide Teilnehmer zugleich bewegen. Das Mobile SCTP kann alternativ auch einen Mobile IP Home Agent verwenden und in diesem Fall die Verbindung zwischen den beiden Endpunkten über diesen aufrechterhalten und so einen Abbruch vermeiden. Bei beiden Ansätzen sind jedoch erhebliche Änderungen

[8] Üblicherweise werden eingehende Verbindungen bis auf wenige bestimmte Ausnahmen aus Sicherheitsgründen sogar aktiv blockiert.

Tabelle 2.2: Vergleich der Ansätze zur Makro-Mobilitätsunterstützung hinsichtlich der Verwendung in heutigen Netzen.

Ansätze	**ISO/OSI Schicht**	**Netzseitige Unterstützung**	**Modifikationen am Terminal**	**Modifikationen bei den Kommunikationspartnern**
Mobile IP ohne RO	3	HA, ggf. FA	Clientprogramm zur Anpassung des Routing	Nicht erforderlich
Mobile IP mit RO	3	HA, ggf. FA	Clientprogramm zur Anpassung des Routing	Clientprogramm zur Anpassung des Routing
Hierarch. Mobile IP	3	HA, MAP bzw. FA	Clientprogramm zur Anpassung des Routing	Nicht erforderlich
HIP, i3	$3\frac{1}{2}$	Server	Zusätzliches Protokoll, Änderung der TCP/UDP Implementierungen, ggf. Änderung von Anwendungen	Zusätzliches Protokoll, Änderung der TCP/UDP Implementierungen, ggf. Änderung von Anwendungen
TCP-Mobilität	4	Dyn. DNS✧	Modifizierte TCP Implementierung, ggf. Änderung von Anwendungen	Modifizierte TCP Implementierung, ggf. Änderung von Anwendungen
Mobile SCTP	4	Dyn. DNS✧ und/oder HA	Neuartige SCTP-Implementierung, Änderung aller Anwendungen	Neuartige SCTP-Implementierung, Änderung aller Anwendungen
MSOCKS	4	Server am Netzzugang	Änderung der TCP/UDP Implementierungen, ggf. Änderung von Anwendungen	Nicht erforderlich
Mobile SIP	5	SIP-Proxy✧	Neue Mobile SIP-Implementierung, ggf. Änderung von Anwendungen	Nicht erforderlich
RMS	5	SIP-Proxy✧	Wie Mobile SIP, zusätzlich RMS Client	Nicht erforderlich

✧ Zur Unterstützung eingehender Verbindungen

Tabelle 2.3: Kategorisierung der Lösungsansätze zur Makro-Mobilitätsunterstützung.

Kategorie	**Ansätze**
Endgeräte-basiert	TCP-Mobilität, Mobile SCTP
Neue Zwischenschicht ($3\frac{1}{2}$)	HIP, i3
Zentralisierte netzseitige Unterstützung	Mobile IP ohne RO, Mobile SIP, RMS
Lokale netzseitige Unterstützung	MSOCKS
Hybrid	Hierarchisches Mobile IP, Mobile IP mit RO

sowohl an den Terminals als auch den Kommunikationspartnern erforderlich, da jeweils ein modifiziertes Transportprotokoll zum Einsatz kommt. Anwendungen müssen für Mobile SCTP und Mobile TCP nur geändert werden, sofern diese direkt IP-Adressen verwenden.

Zu den Protokollen, die auf einer zentralisierten Unterstützung von Netzseite beruhen, zählen das Mobile IP ohne Routenoptimierung (RO) und das Mobile SIP mitsamt RMS. In beiden Fällen sind keine Änderungen bei den Kommunikationspartnern erforderlich. Mobile IP ist für jegliche Anwendung transparent, jedoch werden alle Datagramme über den Home Agent transportiert. In den besuchten Zugangsnetzen ist optional ein Foreign Agent vorgesehen. Das Mobile SIP ist für jede SIP-basierte Anwendung (z. B. *Voice over IP*) transparent und verwendet einen SIP-Proxy für Signalisierungszwecke.

Eine lokale Unterstützung in der Nähe des Netzzugangs ist bei den MSOCKS in Form eines Transportschicht-Proxy vorgesehen, mit dem sich der Teilnehmer von seinem jeweils aktuellen Aufenthaltsort aus verbindet. Dazu benötigt sowohl das Terminal als auch der Proxy eine modifizierte SOCKS [LGL+96]-Implementierung, während am entfernten Kommunikationspartner keine Änderungen erforderlich sind.

Daneben gibt es die beiden hybriden Ansätze, Mobile IP mit Routenoptimierung und Hierarchisches Mobile IP. Diese weisen Eigenschaften mehrerer Kategorien auf. Das Hierarchische Mobile IP verfügt zusätzlich zum Home Agent noch über eine lokale Unterstützung durch zusätzliche hierarchische Foreign Agents. Diese terminieren im Falle von Handovern innerhalb einer Domäne die Mobile IP-Signalisierung des Terminals. Die Anforderungen an die anderen Einheiten im Netz ändern sich dadurch gegenüber regulärem Mobile IP nicht. Im Gegensatz dazu verwendet das Mobile IP mit Routenoptimierung einen Ende-zu-Ende-Ansatz, um die Datagramme direkt zwischen den Kommunikationsendpunkten zu übertragen. Dies erfordert auf allen potentiellen Gegenstellen ein zusätzliches Clientprogramm zur Änderung der Verkehrslenkung. Die Kategorisierung der verschiedenen Ansätze ist in Tab. 2.3 nochmals zusammengefaßt.

2.7.2 Mikro-Mobilitätsunterstützung und deren Kompatibilität

Die Mikro-Mobilitätsunterstützung auf der Sicherungsschicht (z. B. mittels des oben vorgestellten *Cellular IP*) ist mit den oben aufgezählten Verfahren zur Makro-Mobilitätsunterstützung kompatibel. Hierbei wird die Übermittlung von Sicherungsschicht-Nachrichten (z. B. *Ethernet Frames*) anhand der Sicherungsschicht-Adresse (z. B. MAC-Adresse) sichergestellt, so daß IP-Pakete vom Foreign Agent an das mobile Terminal ohne weitere Routingvorgänge zugestellt werden können. Daher wird von *Cellular IP* sowohl Mobile IP im Foreign Agent-Modus als auch mit kollokierter Zustelladresse unterstützt.

Im Gegensatz dazu läßt sich die Mikro-Mobilitätsunterstützung auf der Netzschicht nicht ohne Weiteres mit Methoden der Makro-Mobilitätsunterstützung auf derselbe Schicht vereinen. Unproblematisch ist bei beiden oben vorgestellten Verfahren – *Hawaii* und *Mombasa* – die Kombination mit Makro-Mobilitätsverfahren, bei denen der Teilnehmer Datagramme an eine topologisch korrekte IP-Adresse erhält. Dies ist bei allen oben aufgeführten Verfahren mit Ausnahme von Mobile IP im Foreign Agent-Modus der Fall. Für diese Variante müßte *Hawaii* beziehungsweise *Mombasa* über eine zusätzliche Schnittstelle verfügen, um das Lenken von Datagrammen an mobile Terminals mit fremden IP-Adressen innerhalb der Domäne zu steuern.

2.7.3 Vergleich der Handoverperformanz

In Tab. 2.4 sind die Unterbrechungsdauern[9] der einzelnen Ansätze zur Mobilitätsunterstützung zusammengefaßt dargestellt. Dabei wird die minimale Unterbrechungsdauer der Datagrammzustellung bei einem *make-before-break* Handover einerseits und einem *break-before-make* Handover andererseits verglichen – jedoch ohne die davon unabhängig auftretende Dauer der Detektion eines Netzwechsels.

Man erkennt, daß die meisten Protokolle einen unterbrechungsfreien *make-before-break* Handover unterstützen. Ausnahmen davon sind die beiden TCP-basierten Ansätze (TCP-Mobilität und MSOCKS), da der TCP-Zustandsautomat keine Pakettransporte während Verbindungsaufbau und -änderung vorsieht. Durch eine geschickte Implementierung könnte zumindest der Empfang von bereits ausgesandten Datagrammen eingerichtet und so die Unterbrechungsdauer halbiert werden.

Im Falle eines *break-before-make* Handover gibt es hingegen deutlichere Unterschiede zwischen den Protokollen. Hier lassen sich drei Gruppen von Protokollen unterscheiden: Ansätze mit netzseitiger Umschaltung, Ende-zu-Ende-Ansätze, sowie kombinierte Ansätze die beiderlei Merkmale aufweisen.

Zur ersten Gruppe zählen Mobile IP ohne Routenoptimierung, das Hierarchische Mobile IP, *i3* und MSOCKS. Hier ist die protokollbedingte Unterbrechung abhängig von der aktuellen Entfernung des Terminals zum netzseitigen Umschaltpunkt, also dem Home Agent, Kreuzungs-Foreign

[9] Angenommen wurde ein Download-Szenario.

Tabelle 2.4: Vergleich der protokollbedingten Unterbrechungsdauern bei Handovern.

Ansätze	**ISO/OSI Schicht**	**Make-before-Break Handover**	**Break-before-Make Handover**
Mobile IP ohne RO	3	Unterbrechungsfrei	Paketumlaufzeit vom Terminal über den neuen FA zum HA
Mobile IP mit RO	3	Unterbrechungsfrei	Doppelte Paketumlaufzeit vom Terminal über den HA zur Gegenstelle
Hierarch. Mobile IP	3	Unterbrechungsfrei	Paketumlaufzeit vom Terminal zum Kreuzungs-FA
HIP	$3\frac{1}{2}$	Unterbrechungsfrei	Paketumlaufzeit zwischen den Endpunkten
i3	$3\frac{1}{2}$	Unterbrechungsfrei	Paketumlaufzeit vom Terminal zum *i3*-Server
TCP-Mobilität	4	Einfache bis doppelte Paketumlaufzeit zwischen den Endpunkten (implementierungsabhängig)	Doppelte Paketumlaufzeit zwischen den Endpunkten (Terminal und Server)
Mobile SCTP	4	Unterbrechungsfrei	Paketumlaufzeit zwischen den Endpunkten über den HA
MSOCKS	4	Bis zur doppelten Paketumlaufzeit zwischen mobilem Terminal und Proxy (implementierungsabhängig)	Doppelte Paketumlaufzeit zwischen mobilem Terminal und Proxy
Mobile SIP	5	Unterbrechungsfrei	Paketumlaufzeit zwischen den Gegenstellen
RMS	5	Unterbrechungsfrei	Paketumlaufzeit zwischen den Gegenstellen

Agent, *i3*-Server beziehungsweise MSOCKS-Proxy. Bedingt durch den Dreiwege-Verbindungsaufbau des TCP weist MSOCKS, im Vergleich zu den anderen Ansätze, die doppelte Paketumlaufzeit aus.

Die zweite Gruppe umfaßt das HIP, die TCP-Mobilität sowie Mobile SIP und RMS. Hier hängt die protokollbedingte Unterbrechungsdauer von der Entfernung der beiden Kommunikationspartner ab. Wiederum ergibt sich hier die doppelte Paketumlaufzeit für den TCP-basierten Ansatz.

Schließlich spielen Mobile IP mit Routenoptimierung und das Mobile SCTP eine Sonderrolle. Beide nutzen eine Ende-zu-Ende-Signalisierung unter Einbeziehung des Home Agent. Dementsprechend beläuft sich die Unterbrechungsdauer bei Mobile SCTP auf die Paketumlaufzeit vom Terminal über den Home Agent zur Gegenstelle. Bei Mobile IP erfordert die Prüfung der Legitimität der Aktualisierungsnachricht einen zusätzlichen Paketumlauf, weshalb die doppelte Dauer benötigt wird.

2.8 Zusammenfassung

In diesem Kapitel werden zunächst die Mobilitätsunterstützung in IP-basierten Systemen im Schichtenmodell eingeordnet und der Begriff der Mobilität gegenüber der Portabilität hinsichtlich *Roaming* und *Handover* abgegrenzt. Daran schließt sich eine Betrachtung der Terminalmobilität aus Sicht der Netztopologie an, wobei unter anderem auf die unterschiedliche Verwendung des Begriffs der *Overlaynetze* eingegangen wird. Schließlich leitet die Betrachtung des Handover zur ausführlichen Darstellung des Mobile IP über.

Mobile IP ist der bedeutendste Standard für IP-basierte Makro-Mobilitätsunterstützung und findet auch außerhalb des eigentlichen Internet Verwendung, z. B. beim CDMA2000 [RAB06]. Neben der Grundfunktionalität werden daher auch die wichtigsten der zahlreichen Verbesserungen und Ergänzungen für Mobile IP vorgestellt. Darunter befindet sich auch das Hierarchische Mobile IP, dessen Funktionalität Makro- und Mikro-Mobilitätsunterstützung kombiniert. Dem gegenübergestellt werden drei repräsentative Verfahren zur Mikro-Mobilitätsunterstützung und hinsichtlich ihrer Kompatibilität mit Mobile IP analysiert. Mit *HIP* und *i3* finden zwei ebenfalls unterhalb der Transportschicht angesiedelte innovative Protokolle Erwähnung.

Diese haben jedoch wie auch die Ansätze zur Mobilitätsunterstützung auf der Transportschicht den Nachteil, daß sie zumeist Änderungen der Protokollimplementierung auf Seiten der entfernten Kommunikationspartner erfordern. Da dies ein wesentlicher Aspekt beim Einsatz der Verfahren ist, werden die vorgestellten Protokolle zur Mobilitätsunterstützung hinsichtlich ihrer notwendigen Änderungen an den unterschiedlichen Einheiten ausführlich verglichen, bevor abschließend auf deren Handoverperformanz eingegangen wird.

Bedingt durch das Abstraktionsprinzip der ISO/OSI-Schichten reduzieren sich die Handovervorgänge der physikalischen Schicht und der Sicherungsschicht für die Mobilitätsprotokolle auf den Aspekt der Konnektivität zu einem IP-Netz. Im folgenden Kapitel wird der Handovervorgang des Wireless LAN aus genau dieser Systemsicht näher untersucht.

Kapitel 3

Die Handover-Hysterese bei Wireless LAN

Wireless LAN der Protokollfamilie IEEE 802.11 ist das derzeit am weitesten verbreitete System für den drahtlosen Zugang zu lokalen IP-Netzen. Im Gegensatz zu den Mobilfunksystemen wurde es nicht für den Betrieb mit mobilen Teilnehmern ausgelegt und bietet daher für diese auch keine besondere Unterstützung. In diesem Kapitel werden die Eigenschaften von Wireless LAN im Hinblick auf die für das Mobilitätsmanagement der darüberliegenden Schichten wichtigen Aspekte abstrahiert.

Basierend auf einer idealisierten Modellierung wird die für das Mobilitätsmanagement besonders wichtige Zeit zwischen zwei Handovern hergeleitet, die im folgenden als *Zwischenhandoverzeit* bezeichnet wird. Diese berücksichtigt den in früheren Arbeiten vernachlässigten *Hysterese-Effekt*. Dieser ergibt sich bei der Handoverstrategie von Wireless LAN aus den Schwellwerten der Empfangspegel einerseits sowie den geometrischen Gegebenheiten andererseits. Verglichen mit heutigen Mobilfunksystemen ist dies eine sehr einfach Strategie, da das Terminal erst nach einer neuen Basisstation zu suchen beginnt, nachdem der Signalpegel unter einen bestimmten Schwellwert gefallen ist.

3.1 Handover bei Wireless LAN

Bei Wireless LAN bilden Basisstationen, die an das gleiche IP-Netz angeschlossen sind, ein *Extended Service Set* (ESS), wobei diese Basisstationen jeweils durch den gleichen *Service Set Identifier* (SSID) gekennzeichnet werden. Im Gegensatz zu zellularen Mobilfunksystemen hat das Netz keinen Einfluß darauf, zu welcher Basisstation das mobile Terminal wechselt, auch den Zeitpunkt des Handover entscheidet das mobile Terminal autonom. Das Netz hat nicht einmal Kenntnis darüber, wann und wohin der mobile Teilnehmer als Nächstes wechselt, bevor er sich nicht an einem neuen Netzzugangspunkt angemeldet hat [YIHK02].

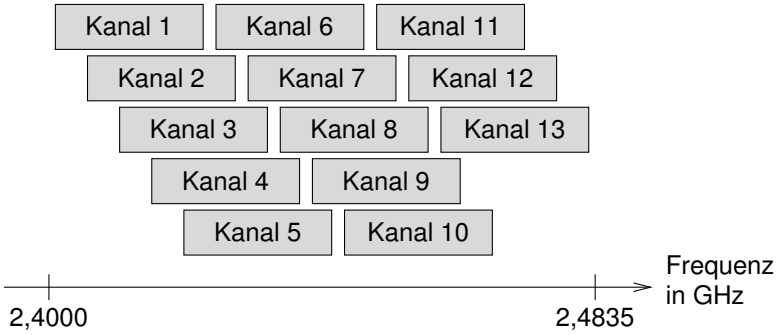


Abbildung 3.1: Kanäle von Wireless LAN 802.11b in Europa.

3.1.1 Kanäle des Wireless LAN

Insgesamt bietet beispielsweise IEEE 802.11b in Europa bis zu 13 Kanäle, wobei die Zahl der Kanäle in manchen Ländern eingeschränkt ist [IEE99c]. Wie in Abb. 3.1 gemäß [IEE99c] dargestellt, überlappen sich die Kanäle bei Wireless LAN (IEEE 802.11b) teilweise. Um die Funkkapazität bestmöglichst zu nutzen, sollten benachbarte Stationen daher nur Nachbarkanäle mit möglichst geringer Interferenz oder – falls möglich – vollkommen überlappungsfreie Kanäle wählen [FAFH04].

Somit ist ein *horizontaler* Handover zwischen zwei Basisstationen des gleichen ESS meist auch mit einem Kanalwechsel verbunden. Insofern könnte ein Terminal, das nur auf einem Kanal gleichzeitig senden oder empfangen kann, auch dann nur Datagramme von einem Netz des aktuellen ESS gleichzeitig empfangen, wenn beim Wireless LAN mehrfache gleichzeitige Assoziationen möglich wären. Daher kann bei der Betrachtung des Handover vereinfachend angenommen werden, daß IP-Pakete (nach der durch den Kanalwechsel bedingten Totzeit) nur von der neuen Basisstation empfangen werden können (siehe Abschnitt 2.1.3).

3.1.2 Verbinden mit einer neuen Basisstation

Bei den heute zumeist verwendeten Systemen wird dann ein Wechsel der Basisstation initiiert, wenn der Empfang der bisherigen nicht mehr ausreichend ist. Dies kann entweder auf zu geringe Signalstärke oder auf Störungen zum Beispiel durch andere Stationen zurückzuführen sein, die auf dem gleichen oder einem überlappenden Nachbarkanal senden. Der Wireless LAN-Netzadapter beginnt dann selbständig nach einer anderen Basisstation zu suchen, die zum gleichen *Extended Service Set* gehört[1]. Wie in Abb. 3.2 gezeigt, wechselt er dazu den Funkkanal[2] und wartet auf *Beacon*-Nachrichten, die von den Basisstationen periodisch gesendet werden. Im Unterschied zu diesem als *passive* Suche bezeichneten Vorgehen kann die mobile Station auch auf dem neuen Kanal einen *Probe Request* senden, der den gesuchten *Extended Service Set*

[1] Es ist auch möglich, Basisstationen unabhängig von der ESSID auszuwählen.

[2] Der Wechsel des Funkkanals darf beispielsweise bei IEEE 802.11 höchstens 224 μs dauern [IEE99a].

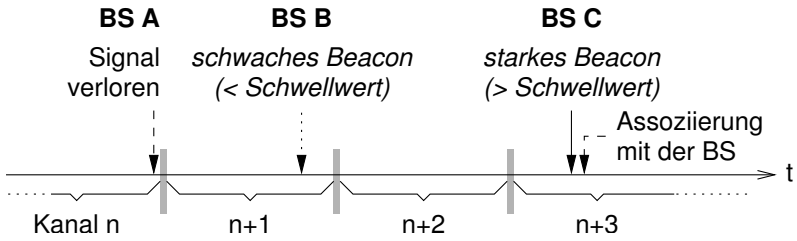


Abbildung 3.2: Beispielhafte Darstellung der Kanalauswahlstrategie von Wireless LAN.

Identifier beinhaltet, was als *aktive* Suche bezeichnet wird. Empfängt eine Basisstation, die zum gleichen *Extended Service Set* gehört, diesen *Probe Request*, so antwortet sie mit einem *Probe Response*. In jedem Falle wird das mobile Terminal eine Basisstation auswählen, deren Empfangspegel oberhalb eines Schwellwertes liegt. Ansonsten wird der Vorgang auf einem anderen Kanal wiederholt. Um den Vorgang zu beschleunigen kann das Terminal auch eine selektive Suche durchführen, also die Kanäle – einer Heuristik folgend – in einer anderen Reihenfolge absuchen [SFRS04, MSA03].

Im Gegensatz zum *Global System for Mobile Communications* (GSM) [EVB01] wird jedoch bei einem Handover immer nur ein Kanal nach dem anderen daraufhin untersucht, ob eine Basisstation empfangbar ist. Es erfolgt dabei kein direkter Vergleich aller empfangbarer Basisstationen mit anschließender Auswahl des stärksten Senders. Daher kann es also unter ungünstigen Empfangsbedingungen geschehen, daß zunächst mehrere aufeinanderfolgende Handover erfolgen bevor schließlich eine empfangsstarke Basisstation ausgewählt wird. Auch werden sogenannte *Ping-Pong-Handover* nicht aktiv vermieden.

Wurde schließlich eine passende Basisstation gefunden, folgen gegenseitige Authentifizierung und schließlich die feste *Assoziierung* des mobilen Terminals mit der Basisstation, wie es in Abb. 3.3 dargestellt ist. Erst damit ist der Handover auf der Sicherungsschicht abgeschlossen, Teilnehmer und Basisstation können nun IP-Pakete miteinander austauschen [IEE99a].

Eine Folge dieser Strategie der Kanalauswahl ist, daß ein mobiles Terminal, welches sich durch die Versorgungsgebiete verschiedener Basisstationen bewegt, erst dann eine neue Basisstation auswählt, wenn der Signalpegel der vorherigen Station unter einen bestimmen Schwellwert gefallen ist. Auf diese Weise entsteht eine *geometrische Hysterese*, da es von der geometrischen Anordnung der vorhergehenden Basisstation abhängt, wann sich ein Teilnehmer mit der neuen Basisstation verbindet. Mit welcher Basisstation ein Teilnehmer aktuell assoziiert ist, hängt auch hier nicht nur von den aktuellen Signalpegeln ab, sondern auch von dem von ihm beschrittenen Weg. Bewegt sich der Teilnehmer auf dem gleichen Weg in der umgekehrten Richtung, so finden die Handover nicht am gleichen Ort statt, so daß sich die in Abb. 3.4 beispielhaft aufgetragene Zuordnung ergibt.

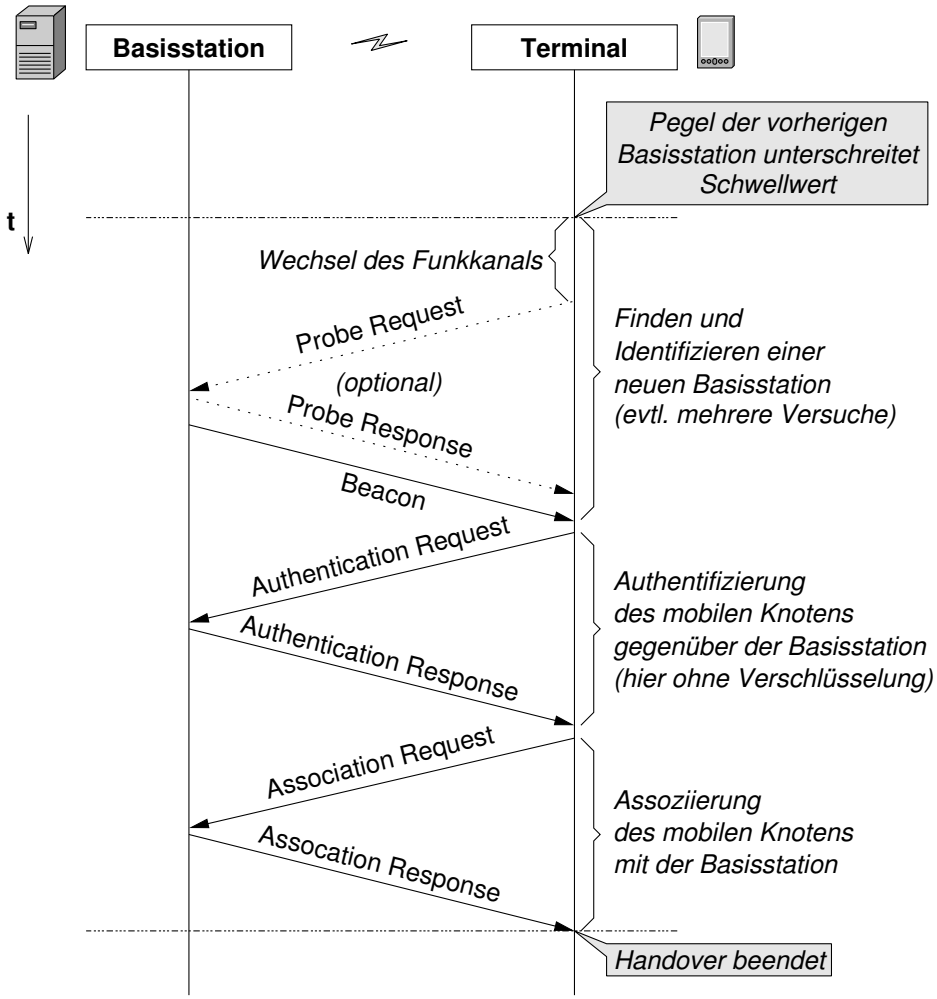


Abbildung 3.3: Nachrichtenflußdiagramm eines Handover bei Wireless LAN (IEEE 802.11).

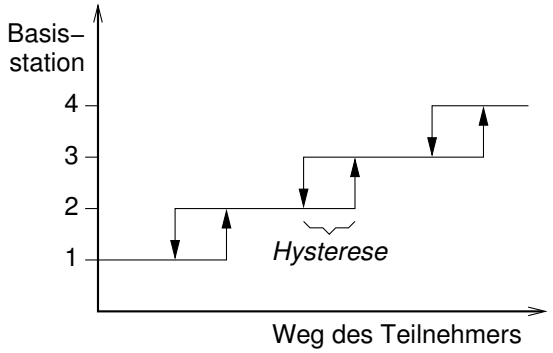


Abbildung 3.4: Hysterese-Effekt beim Wechsel zwischen Wireless LAN-Basisstationen.

3.2 Annahmen für die analytische Modellierung

Im folgenden wird nun hergeleitet, wie sich die Eigenschaften der Kanalauswahl und der daraus resultierenden geometrischen Hysterese auf die Dauer zwischen zwei Handovern auswirkt. Dabei werden die in [**TBH05**] publizierten Ergebnisse erweitert.

3.2.1 Funkwellenausbreitung

Bei Wireless LAN, z. B. nach dem Standard IEEE 802.11a, wird die Kanalkodierung dynamisch an die aktuellen Eigenschaften des Funkkanals angepaßt. Unter der Annahme homogener Funkwellenausbreitung ohne störende Sender ist die Funkreichweite lediglich durch die Sendeleistung P_{TX} und die Empfindlichkeit des Empfängers begrenzt. Entsprechend dem logarithmischen Pfadverlustmodell ist der Empfangspegel P_{RX} beim Empfänger in Abhängigkeit vom Verlust P_{loss} nach [LR96] gegeben durch

$$P_{\mathrm{RX}} = P_{\mathrm{TX}} - P_{\mathrm{loss}} \qquad \text{mit} \quad P_{\mathrm{loss}} = 10 \log \left(\left(\frac{4\pi f}{\mathrm{c}} \right)^2 d^{\gamma} \right) \tag{3.1}$$

Dabei ist d die Entfernung zwischen Sender und Empfänger, γ der Dämpfungskoeffizient und c die Lichtgeschwindigkeit. Die Pegel P_{RX} und P_{TX} sind in dBm angegeben.

Entfernt sich nun das Terminal von der Basisstation, wird mit sinkendem Empfangspegel schrittweise auch die nominale Datenrate reduziert, um den effektiven Datendurchsatz zu maximieren. In Abb. 3.5 ist die nominelle Datenrate für den Standard IEEE 802.11a sowie ein standardkonformes Produkt aufgetragen.

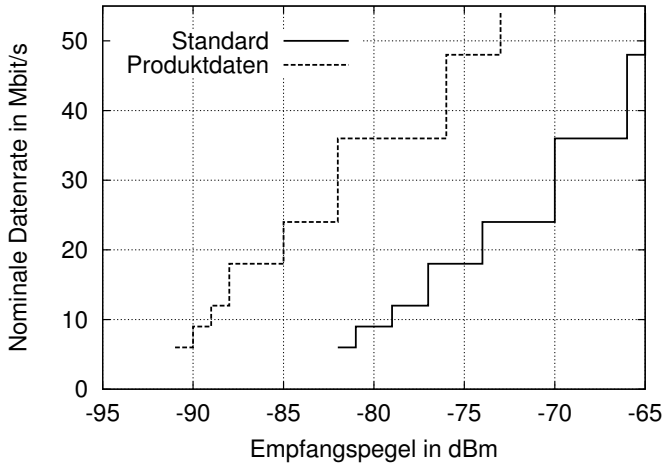


Abbildung 3.5: Nominale Datenrate bei verschiedenen Empfangspegeln.

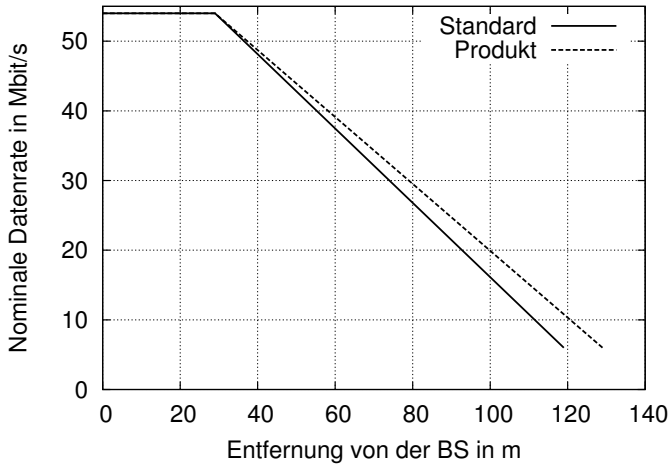


Abbildung 3.6: Linear approximierte Funktion der nominalen Datenrate in Abhängigkeit von der Entfernung.

Bei homogener, störungsfreier Funkwellenausbreitung läßt sich der Empfangspegel direkt in eine Entfernung von der Basisstation umrechnen, wie in Abb. 3.6 gezeigt. Dabei wurde ein Dämpfungskoeffizient von γ = 2,8 angenommen. Dies ist der Mittelwert zwischen der Dämpfung im freien Raum mit γ = 2 und in einer typischen Büroumgebung mit γ = 3,6 [SPG+03]. Sendeleistung und Empfängerempfindlichkeit entsprechen den für eine Sendefrequenz von 5,3 GHz in [IEE99b, Net04] jeweils gegebenen Werten.

Da sich die Datenrate mit zunehmender Entfernung von der Basisstation reduziert, belegen weiter entfernte Teilnehmer für die Übertragung der gleichen Datenmenge den Funkkanal entsprechend länger. Dies reduziert natürlich die für andere Stationen zur Verfügung stehende Kapazität. Daher ist es bei Wireless LAN vorgesehen, daß bei der Basisstation auch eine minimal zulässige Datenrate eingestellt werden kann. Kann diese nicht erreicht werden, muß der Teilnehmer auf eine andere Basisstation wechseln.

Ein Handover wird also immer dann veranlaßt, wenn entweder der *Signalpegel* einen bestimmten Schwellwert unterschreitet oder wenn die erforderliche *minimale Datenrate* nicht erreicht werden kann.

3.2.2 Modellannahmen bezüglich der Topologie

Ein System von Basisstationen wird normalerweise so ausgelegt sein, daß überall eine zulässige Basisstation zur Verfügung steht. Um analytisch diejenige Zeitdauer bestimmen zu können, die ein Teilnehmer mit einer Basisstation assoziiert ist, muß zunächst die Länge seiner entsprechenden Wegstrecke durch das Versorgungsgebiet einer Basisstation bestimmt werden. Dieses wird als Kreisscheibe modelliert, wobei der Radius R durch Sendestärke, Dämpfung und den mindestens erforderlichen Empfangspegel bestimmt wird, wie oben dargelegt. Die Teilnehmer

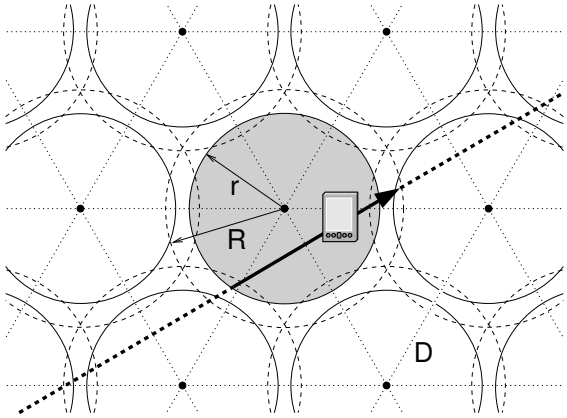


Abbildung 3.7: Idealisierte regelmäßige Zelltopologie mit homogenen Bedingungen.

bewegen sich in einer Ebene mit homogener Funkwellenausbreitung, die vollständig von den Basisstationen abgedeckt wird. Die Basisstationen sind entlang eines äquidistanten hexagonalen Gitters mit dem Zellabstand D angeordnet, so daß jede sechs Nachbarn hat, wie in Abb. 3.7 dargestellt. Jede Basisstation hat eine maximale Reichweite R (gestrichelte Linie), bei der die Teilnehmer den Funkkontakt verlieren. Die Teilnehmer verbinden sich dann umgehend mit der neuen Basisstation. Die Punkte, an denen sich die aus allen Richtungen kommenden Teilnehmer mit einer Basisstation verbinden, ergeben einen Hypotrochoid. Dieser läßt sich mittels des durchgezogenen Kreises mit Radius r $(< R)$ annähern.

Damit die Hysterese überhaupt in eindeutig berechenbarer Form in Erscheinung tritt, muß es in jeder Zelle einen Bereich um die Basisstation geben, der nicht von anderen Zellen abgedeckt wird. Dort wird sich daher jeder Teilnehmer mit dieser einen Basisstation assoziieren. Folglich muß für die Reichweite R einer Basisstation gelten

$$\frac{1}{\sqrt{3}}D \leq R < D \tag{3.2}$$

Je größer R und somit auch die Überlappung ist, desto mehr Mobilstationen verbinden sich im Kreisring zwischen dem *äußeren* Radius R und dem äquivalenten *inneren* Radius r mit der Basisstation. Eine gewisse Überlappung ist dabei unumgänglich, um Robustheit gegenüber den hier nicht weiter betrachteten Störeinflüssen wie beispielsweise Abschattungen zu erhalten. Eine größere Überlappung bewirkt dabei eine größere geometrische Hysterese, im Gegenzug tritt die geometrische Hysterese jedoch bei einem geringeren Anteil von Mobilteilnehmern in Erscheinung. Eine zu große Überlappung sollte aus Gründen der Wirtschaftlichkeit wie auch der Performanz vermieden werden, nicht zuletzt da größere Überlappungen der Versorgungsgebiete stets auch größere Gleichkanalstörungen anderer Zellen verursachen.

Für die Bewegung der mobilen Teilnehmer wird das *Random Direction*-Modell [Bet01] angenommen. Entsprechend diesem Modell bewegt sich jedes Terminal mit einer zufällig gewählten, konstanten Geschwindigkeit in eine zufällige – jedoch ebenfalls konstante – Richtung. In

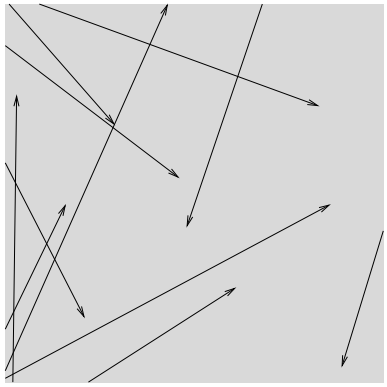


Abbildung 3.8: Zufällige Trajektorien von Knoten im *Random Direction*-Mobilitätsmodell.

Abb. 3.8 sind beispielhaft die Trajektorien einiger Knoten dargestellt. Die Geschwindigkeiten v und die Richtungen ϕ sind gleichverteilt entsprechend

$$f_v(v) = \frac{1}{v_{\max} - v_{\min}} \quad \text{für} \quad v_{\min} \leq v \leq v_{\max} \tag{3.3}$$

$$f_\phi(\phi) = \frac{1}{2\pi} \quad \text{für} \quad 0 \leq \phi \leq 2\pi \tag{3.4}$$

3.2.3 Berechnung des äquivalenten inneren Radius

Entsprechend seiner Definition beschreibt der äquivalente innere Radius r eine kreisförmige Approximation der Fläche, die ausschließlich von der betreffenden Basisstation versorgt wird. Der äquivalente innere Radius r ergibt sich also aus der exklusiv abgedeckten Fläche A_1 als

$$r = \sqrt{\frac{A_1}{\pi}} \quad \text{mit} \quad A_1 = R^2\pi - 6(A_2 - A_3) \tag{3.5}$$

wobei A_2 und A_3 die Schnittflächen zweier beziehungsweise dreier benachbarter Zellen bezeichnen. Diese lassen sich aus dem Zellabstand D und der Reichweite R der Basisstationen berechnen als

$$A_2 = 2R^2 \arccos\left(\frac{D}{2R}\right) - \frac{D}{2}\sqrt{4R^2 - D^2} \tag{3.6}$$

$$A_3 = \frac{\sqrt{3}}{4}\zeta^2 + 3R^2 \arcsin\left(\frac{\zeta}{2R}\right) - \frac{3}{2}\zeta\sqrt{R^2 - \frac{\zeta^2}{4}} \tag{3.7}$$

$$\text{mit der Hilfsgröße} \quad \zeta = \sqrt{3R^2 - \frac{3}{4}D^2} - \frac{1}{2}D$$

Schließlich erhält man die Funktion für den äquivalenten inneren Radius r, die in Abb. 3.9 aufgetragen ist. Ebenfalls aufgetragen ist die Gerade zwischen der bei dieser Modellierung minimalen

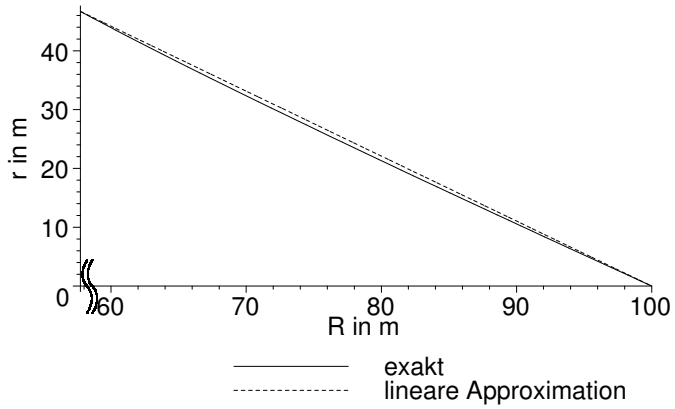


Abbildung 3.9: Die Funktion $r(R)$ für D = 100 m.

und maximalen Reichweite der Basisstation gemäß Gl. 3.2, die im folgenden vereinfachend verwendet wird:

$$r(R) \approx (D-R)\frac{\sqrt{\frac{3\sqrt{3}}{\pi}-1}}{\sqrt{3}-1} \approx 1{,}105\,(D-R) \tag{3.8}$$

3.2.4 Wegstrecke zwischen zwei Handovern

Die Strecke Z_h, die der mobile Teilnehmer zwischen zwei aufeinanderfolgenden Handovern zurücklegt, hängt vom Winkel α ab wie in Abb. 3.10 dargestellt. Dabei wird angenommen, daß sich der Teilnehmer im Abstand des Radius r von der Basisstation mit dieser verbindet und sie erst im Abstand des Radius R wieder verläßt. Durch geometrische Betrachtungen ergibt sich folgende Beziehung zwischen der Wegstrecke Z_h, den beiden Radien r und R sowie dem Winkel α:

$$Z_h(\alpha) = r\cos\alpha + \sqrt{R^2 - r^2\sin^2\alpha} \tag{3.9}$$

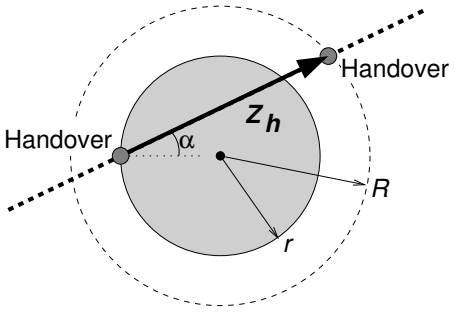


Abbildung 3.10: Wegstrecke Z_h eines Knotens zwischen zwei Handovern.

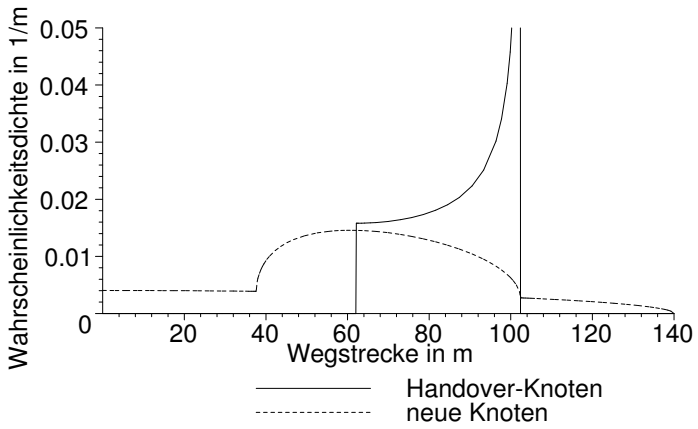


Abbildung 3.11: Verteilung der Pfadlängen von Handoverknoten und neuen Knoten.

Da für das Berechnen der Wahrscheinlichkeitsdichtefunktion eine umkehrbare Funktion erforderlich ist, wird im folgenden eine Näherung der Form

$$Z_h(\alpha) \approx b \cos \alpha + c \tag{3.10}$$

mit den beiden Hilfsvariablen b, c verwandt. Setzt man die ursprüngliche Gleichung und die Approximation für die Werte $\alpha = 0$ und $\alpha = \frac{\pi}{2}$ gleich, so erhält man zwei Gleichungen aus denen sich die Hilfsvariablen b und c bestimmen lassen als

$$b = r + R - \sqrt{R^2 - r^2} \tag{3.11}$$
$$c = \sqrt{R^2 - r^2} \tag{3.12}$$

Über die Inversion der Näherung für $Z_h(\alpha)$ erhält man analog zu [HR86] die Wahrscheinlichkeitsdichtefunktion von Z_h als

$$f_{Zh}(z) = f_\alpha(\alpha) \left| \frac{\mathrm{d}\alpha(z)}{\mathrm{d}z} \right| = \frac{2}{\pi} \frac{1}{\sqrt{b^2 - (z-c)^2}} \qquad \text{für} \quad c \le z \le b + c \tag{3.13}$$

wenn man aus Gl. 3.4 eine Gleichverteilung der Winkel α in $[0..\frac{\pi}{2}]$ entsprechend dem *Random Direction*-Mobilitätsmodell annimmt.

In Abb. 3.11 ist diese Wahrscheinlichkeitsdichtefunktion der Wegstrecken von Handoverknoten dargestellt. Zum Vergleich ist auch die Wahrscheinlichkeitsdichtefunktion neuer Knoten aufgetragen, die im Anhang A.1 berechnet wird. Für den Graphen wurde der Zellabstand D = 100 m und die Reichweite R = 70 m angenommen, woraus sich ein innerer äquivalenter Radius r = 32,35 m ergibt. Für den Wert $Z = R + r$ – also eine exakt mittige Querung der Zelle – strebt die Wahrscheinlichkeitsdichtefunktion gegen Unendlich.

3.3 Handoverzeiten

Aus der oben hergeleiteten Verteilung der Wegstrecken zwischen zwei Handovern läßt sich mit Kenntnis der Geschwindigkeiten der Knoten die Dauer zwischen zwei Handovern ermitteln. Entsprechend dem *Random Direction*-Modell haben die Teilnehmer eine konstante Geschwindigkeit, die im Intervall $[v_{\min}..v_{\max}]$ gleichverteilt ist. Zunächst werde $v_{\min} \neq 0$ angenommen.

Nach [BHPC04] kann die Wahrscheinlichkeitsdichtefunktion der Zeit $f_T(t)$ in Abhängigkeit von der Wahrscheinlichkeitsdichtefunktion der Strecke $f_Z(Z)$ sowie der Wahrscheinlichkeitsdichtefunktion der Geschwindigkeit $f_v(v)$ bestimmt werden als

$$f_T(t) = \int_{v_{\min}}^{v_{\max}} v\, f_Z(vt)\, f_v(v)\, \mathrm{d}v \tag{3.14}$$

Damit wird zunächst die *allgemeine* Wahrscheinlichkeitsdichtefunktion der Zwischenhandoverzeiten bestimmt, bei der alle Knoten in gleicher Weise berücksichtigt werden. Anschließend wird zusätzlich in Betracht gezogen, daß schnellere Knoten häufiger einen Handover durchführen [XG93] und die sich daraus ergebenden Funktionen miteinander verglichen.

3.3.1 Allgemeine Zwischenhandoverzeit

Setzt man Gl.en (3.13) und (3.3) in Gl. (3.14) ein, läßt sich die Wahrscheinlichkeitsdichtefunktion der allgemeinen Zwischenhandoverzeit herleiten als

$$f_{Th}(t) = \frac{2}{\pi}\frac{1}{\Delta v}\int_{v_{\min}}^{v_{\max}} \frac{v}{\sqrt{b^2-(vt-c)^2}}\,\mathrm{d}v \tag{3.15}$$

mit $c \leq vt \leq b+c$ und $\Delta v = v_{\max} - v_{\min}$ sowie $b = r + R - \sqrt{R^2-r^2}$, $c = \sqrt{R^2-r^2}$ wie oben definiert. Wendet man nun die Substitution $\gamma = vt$ (für $c \leq \gamma \leq b+c$) an, erhält man

$$f_{Th}(t) \;=\; \frac{2}{\pi}\frac{1}{\Delta v}\frac{1}{t^2}\int_{v_{\min}t}^{v_{\max}t} \frac{\gamma}{\sqrt{b^2-(\gamma-c)^2}}\,\mathrm{d}\gamma \tag{3.16}$$

Bei der Lösung dieses Integrals müssen drei Fälle unterschieden werden aufgrund der sich in Gl. (3.13) zusätzlich ergebenden Randbedingungen:

$$c \;\leq\; v_{\min}\cdot t \tag{3.17}$$

$$v_{\max}\cdot t \;\leq\; b+c \tag{3.18}$$

Diese Bedingungen sind in Abb. 3.12 über der t-x-Ebene dargestellt. Dabei wird angenommen, daß $v_{\min}$ und $v_{\max}$ so weit auseinander liegen, daß folgende Bedingung eingehalten wird:

$$\frac{b+c}{v_{\max}} \leq \frac{c}{v_{\min}} \qquad \text{oder} \qquad \frac{v_{\min}}{v_{\max}} \leq \frac{\sqrt{R^2-r^2}}{R+r} \tag{3.19}$$

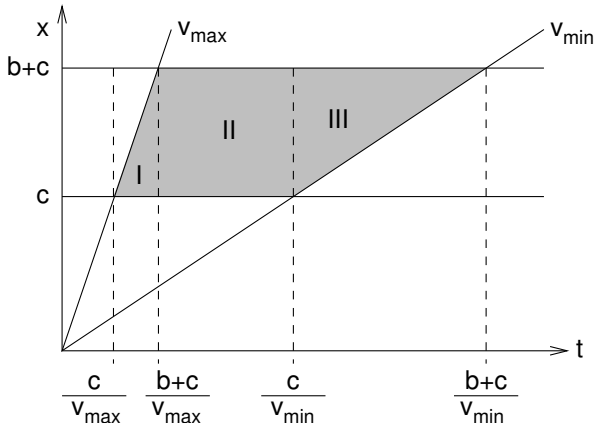


Abbildung 3.12: Randbedingungen für v in der t-x-Ebene.

Es lassen sich dann die drei Fälle wie folgt angeben:

$$\begin{aligned} \text{Fall } \mathbf{I}\text{:} \quad & \frac{c}{v_{\max}} \leq t \leq \frac{b+c}{v_{\max}} \\ \text{Fall } \mathbf{II}\text{:} \quad & \frac{b+c}{v_{\max}} < t \leq \frac{c}{v_{\min}} \\ \text{Fall } \mathbf{III}\text{:} \quad & \frac{c}{v_{\min}} < t \leq \frac{b+c}{v_{\min}} \end{aligned} \tag{3.20}$$

Löst man nun das Integral für jeden dieser Fälle, so ergibt sich für die Wahrscheinlichkeitsdichtefunktion der allgemeinen Zwischenhandoverzeit schließlich:

$$f_{Th}(t) = \begin{cases} \frac{2}{\pi}\frac{1}{\Delta v}\frac{1}{t^2}\left(b - \sqrt{b^2 - (v_{\max}t - c)^2} - c \cdot \arcsin\left(\frac{c - v_{\max}t}{b}\right)\right) & \text{für Fall I} \\ \frac{2}{\pi}\frac{1}{\Delta v}\frac{1}{t^2}\left(\frac{\pi}{2}c + b\right) & \text{für Fall II} \\ \frac{2}{\pi}\frac{1}{\Delta v}\frac{1}{t^2}\left(\sqrt{b^2 - (v_{\min}t - c)^2} + c \cdot \arccos\left(\frac{c - v_{\min}t}{b}\right)\right) & \text{für Fall III} \end{cases} \tag{3.21}$$

Diese ist in Abb. 3.13 für die Werte D = 100 m, R = 70 m, $v_{\min}$ = 1,4 $\frac{\text{m}}{\text{s}}$, und $v_{\max}$ = 14 $\frac{\text{m}}{\text{s}}$ gezeichnet. Wie erwartet gibt es eine bestimmte minimale Zwischenhandoverzeit, da die Hysterese eine minimale Wegstrecke bedingt und hier eine endliche Maximalgeschwindigkeit angenommen wurde. Ebenso gibt es eine maximale Dauer, da eine Minimalgeschwindigkeit $v_{\min} > 0$ vorausgesetzt wurde. Bei einer Minimalgeschwindigkeit $v_{\min} = 0$, wenn also auch unbewegte Teilnehmer betrachtet werden, strebt der Erwartungswert der Zwischenhandoverzeit gegen Unendlich.

3.3.2 Verzerrte Verteilung der Zwischenhandoverzeit

Das *Random Direction*-Modell für die Bewegung der Teilnehmer, welches hier vorausgesetzt wurde, nimmt eine Gleichverteilung sowohl der Richtungen als auch der Geschwindigkeiten

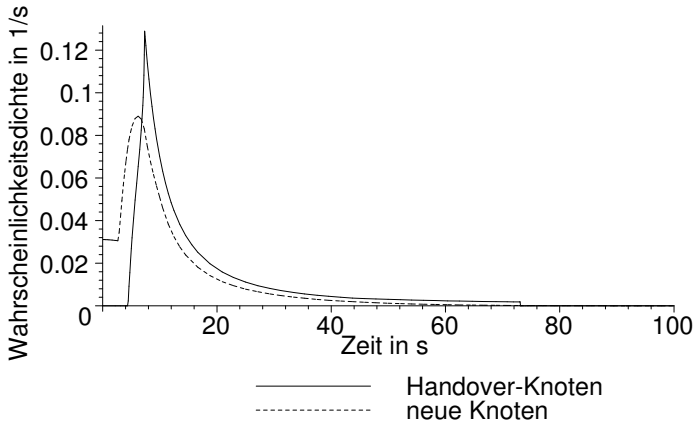


Abbildung 3.13: Zeit zum ersten beziehungsweise nächsten Handover für neue Knoten und Handoverknoten im allgemeinen Fall.

an. Letzteres trifft jedoch dann nicht zu, wenn ausschließlich diejenigen Teilnehmer betrachtet werden, die eine Zelle nach einem Handover betreten. Dies ist darauf zurückzuführen, daß Terminals mit einer geringeren Geschwindigkeit nur mit niedrigerer Wahrscheinlichkeit eine Zellgrenze überschreiten als schnellere Terminals. Diese als *Biased Sampling Problem* bezeichnete Eigenschaft führt zu einer *Verzerrung* der Wahrscheinlichkeitsdichtefunktion der an sich gleichverteilten Geschwindigkeiten.

Die – mit einem Stern markierte – verzerrte Wahrscheinlichkeitsdichtefunktion der Geschwindigkeiten derjenigen Teilnehmer, die mindestens einen Handover ausgeführt haben, läßt sich nach [XG93] angeben als

$$f_v^\star(v) = \frac{v\, f_v(v)}{\mu_v} \tag{3.22}$$

wobei $f_v(v)$ die allgemeine Geschwindigkeitsverteilung und $\mu_v = \int_0^\infty v\, f_v(v)\,\mathrm{d}v$ der zugehörige Erwartungswert ist. Für den Fall einer Gleichverteilung der Geschwindigkeiten resultiert daraus die Verteilung

$$f_v^\star(v) = \frac{2v}{v_{\max}^2 - v_{\min}^2} \qquad \text{für} \quad v_{\min} \le v \le v_{\max} \tag{3.23}$$

Die verzerrte Wahrscheinlichkeitsdichtefunktion der Zwischenhandoverzeit läßt sich unter Verwendung von Gl. (3.14) berechnen als

$$f_{Th}^\star(t) = \underbrace{\frac{2}{\pi}\frac{2}{v_{\max}^2 - v_{\min}^2}}_{\eta} \int_{v_{\min}}^{v_{\max}} \frac{v^2}{\sqrt{b^2 - (vt-c)^2}}\,\mathrm{d}v \qquad \text{mit} \quad c \le vt \le b + c \tag{3.24}$$

Wiederholt man die gleichen Schritte wie zuvor für die allgemeine Wahrscheinlichkeitsdichtefunktion nun für die in Gl. (3.20) gegebenen Fälle, ergibt sich für die verzerrte Zwischenhand-

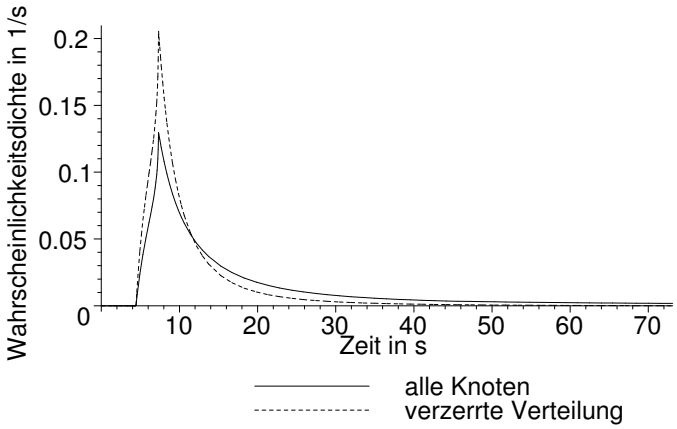


Abbildung 3.14: Allgemeine und verzerrte Zwischenhandoverzeit.

overzeit schließlich folgender Zusammenhang, dessen Graph in Abb. 3.14 ersichtlich ist:

$$f^{\star}_{Th}(t) = \begin{cases} \frac{\eta}{t^3}\left(\frac{2c^2+b^2}{2}\arcsin\left(\frac{v_{\max}t-c}{b}\right) - \right. & \\ \qquad \left. -\frac{3c+v_{\max}t}{2}\sqrt{b^2-(v_{\max}t-c)^2}+2bc\right) & \text{für Fall I} \\ \frac{\eta}{t^3}\left(\frac{\pi}{2}\frac{2c^2+b^2}{2}+2bc\right) & \text{für Fall II} \\ \frac{\eta}{t^3}\left(\frac{2c^2+b^2}{2}\arccos\left(\frac{v_{\min}t-c}{b}\right)+\right. & \\ \qquad \left. +\frac{3c+v_{\min}t}{2}\sqrt{b^2-(v_{\min}t-c)^2}\right) & \text{für Fall III} \end{cases} \tag{3.25}$$

mit $\eta = \dfrac{2}{\pi}\dfrac{2}{v_{\max}^2 - v_{\min}^2}$, $b = r + R - \sqrt{R^2-r^2}$, $c = \sqrt{R^2-r^2}$ wie oben definiert.

Wie man in Abb. 3.14 erkennt, reduziert die Verzerrung der Geschwindigkeitsverteilung die Divergenz der Zwischenhandoverzeiten.

Da bei der verzerrten Verteilung berücksichtigt wird, daß unbewegte Teilnehmer keine Handover durchführen, kann in Gl. (3.25) auch eine Minimalgeschwindigkeit von $v_{\min} = 0$ eingesetzt werden. Im Gegensatz zum obigen allgemeinen Fall ergibt sich hier auch ein endlicher Erwartungswert. In obiger Gleichung gilt dann Fall II bis Unendlich und Fall III entfällt dementsprechend, so daß sich folgende vereinfachte Wahrscheinlichkeitsdichtefunktion der verzerrten Zwischenhandoverzeit ergibt:

$$f^{\star}_{Th0}(t) = \begin{cases} \frac{4}{v_{\max}^2 t^3 \pi}\left(\frac{2c^2+b^2}{2}\arcsin\left(\frac{v_{\max}t-c}{b}\right) - \right. & \\ \qquad \left. -\frac{3c+v_{\max}t}{2}\sqrt{b^2-(v_{\max}t-c)^2}+2bc\right) & \text{für Fall I} \\ \frac{4}{v_{\max}^2 t^3 \pi}\left(\frac{\pi}{2}\frac{2c^2+b^2}{2}+2bc\right) & \text{für } \frac{b+c}{v_{\max}} < t < \infty \end{cases} \tag{3.26}$$

3.4 Evaluierung

3.4.1 Zwischenhandoverzeit und Zellaufenthaltsdauer

Wie eingangs beschrieben, stellt die in früheren Publikationen betrachtete Aufenthaltsdauer innerhalb einer Zelle (Zellaufenthaltsdauer, *Cell Dwell Time*) lediglich die Zeit zwischen dem Betreten und dem Verlassen des Versorgungsgebietes dar. Darüber hinausgehende Eigenschaften von Wireless LAN, wie beispielsweise die durch die Strategie der Kanalselektion hervorgerufene Hysterese bei Handovern, werden dabei jedoch außer acht gelassen. Setzt man in Gl. (3.24) den äquivalenten inneren Radius r gleich der Reichweite R (somit $b = 2R$, $c = 0$) und eliminiert auf diese Weise jegliche Hysterese, erhält man die bloße Zellaufenthaltsdauer:

$$f^{\star}_{Dh}(t) = \underbrace{\frac{2}{\pi}\frac{2}{v_{\max}^2 - v_{\min}^2}}_{\eta} \int_{v_{\min}}^{v_{\max}} \frac{v^2}{\sqrt{4R^2 - (vt)^2}}\,\mathrm{d}v \qquad \text{mit} \quad c \le vt \le b + c \tag{3.27}$$

Da jedoch mit $r = R$ die Annahmen in Gl. (3.19) nicht erfüllt werden können, ändert sich die in Abb. 3.12 gezeigte Fallunterscheidung für die Integration. Es ergibt sich schließlich die gleiche Beziehung wie in [JB00]:

$$f^{\star}_{Dh}(t) = \begin{cases} \frac{\eta}{t^3}\left(\frac{v_{\min}t}{2}\sqrt{4R^2 - v_{\min}^2t^2} - \frac{v_{\max}t}{2}\sqrt{4R^2 - v_{\max}^2t^2} + \right. & \\ \qquad \left. +2R^2\arcsin\left(\frac{v_{\max}t}{2R}\right) - 2R^2\arcsin\left(\frac{v_{\min}t}{2R}\right)\right) & \text{für} \quad 0 < t < \frac{2R}{v_{\max}} \\ \frac{\eta}{t^3}\left(\frac{v_{\min}t}{2}\sqrt{4R^2 - v_{\min}^2t^2} + R^2\pi - 2R^2\arcsin\left(\frac{v_{\min}t}{2R}\right)\right) & \text{für} \quad \frac{2R}{v_{\max}} < t < \frac{2R}{v_{\min}} \end{cases} \tag{3.28}$$

mit $\eta = \frac{2}{\pi}\frac{2}{v_{\max}^2 - v_{\min}^2}$ wie oben definiert.

Wie man in Abb. 3.15 und 3.16 sieht, gibt es im Fall der bislang verwendeten Aufenthaltsdauer auch eine gewisse Wahrscheinlichkeit für sehr kurze Aufenthaltsdauern, die Wahrscheinlichkeitsdichtefunktion hat auch für die Zeitdauer "Null" einen Wert größer Null. Im Gegensatz dazu läßt die bei der Zwischenhandoverzeit beinhaltete Hysterese von Wireless LAN genau dies nicht zu. Hier gibt es eine durch die Größe der Hysterese und die angenommene Maximalgeschwindigkeit bestimmte minimale Dauer zwischen zwei Handovern, kürzere Zeiten kommen nicht vor.

3.4.2 Abgebrochene Handover

Ein wichtiger Aspekt bei der Analyse drahtloser Zugangsnetze ist die Wahrscheinlichkeit eines unvollständigen beziehungsweise abgebrochenen Handover (P_{abbr}). Verläßt ein Teilnehmer das neue Netz noch bevor die Handoverprozedur beendet ist, so hat er lediglich Signalisierungs-, aber keine Nutzdaten übertragen. Ein Abbruch eines Handover liegt immer dann vor, wenn die Zwischenhandoverzeit kürzer ist als die Dauer t_{MIP} der mit dem Handover verbundenen Signalisierungsprozeduren, also beispielsweise der Mobile IP-Registrierung. Mit Hilfe der inversen Verteilungsfunktion der Mobile IP-Handoverdauern $\bar{F}_{\mathrm{MIP}}(t) = \mathrm{P}(t < t_{\mathrm{MIP}})$, die auf den Messungen

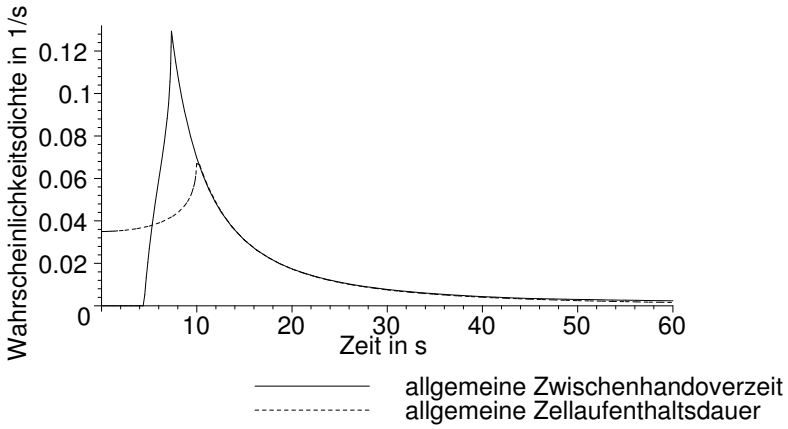


Abbildung 3.15: Zwischenhandoverzeit und Zellaufenthaltsdauer (jeweils allgemeine Geschwindigkeitsverteilung).

in Abschnitt 6.4.3 beruht, läßt sich die Wahrscheinlichkeit eines Handoverabbruchs entsprechend berechnen als

$$P_{\mathrm{abbr}} = \int_0^\infty f_\tau(t)\, \bar{F}_{\mathrm{MIP}}(t)\, \mathrm{d}t \tag{3.29}$$

wobei $f_\tau(t)$ die jeweils angenommene Verteilung der Zeiten zwischen zwei Handovern ist. Unter Verwendung der *Cell Dwell Time* aus Gl. (3.28) erhält man für die oben gewählten Werte (D = 100 m und R = 70 m) eine Handoverabbruchwahrscheinlichkeit von $P_{\mathrm{abbr}} = 12,5\%$.

Berücksichtigt man jedoch die Hysterese, indem stattdessen die in Gl. (3.25) hergeleitete Zwischenhandoverzeit verwendet wird, ergeben sich bei den gleichen Werten keine Handoverabbrüche, die auf das vorzeitige Verlassen der Funkzelle zurückzuführen wären. Allerdings können Handoverabbrüche aufgrund der eingangs erwähnten Störeffekte (Beschränkung der Modellie-

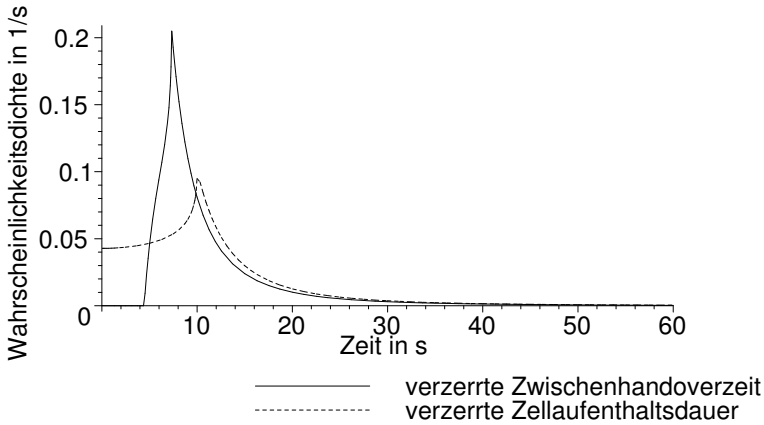


Abbildung 3.16: Zwischenhandoverzeit und Zellaufenthaltsdauer (jeweils verzerrte Geschwindigkeitsverteilung).

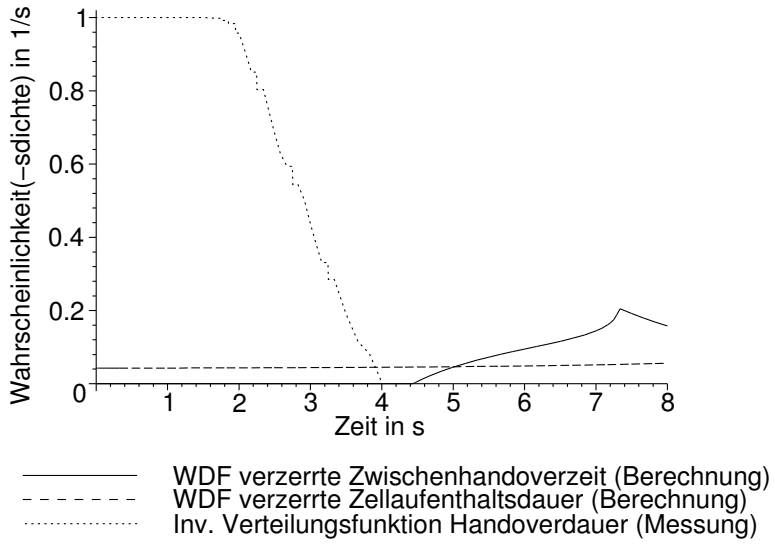


Abbildung 3.17: Analytisch ermittelte Wahrscheinlichkeitsdichtefunktion (WDF) der Zwischenhandoverzeit beziehungsweise Zellaufenthaltsdauer und die inverse Verteilungsfunktion der gemessenen Dauern der Mobile IP Registrierungsprozedur.

rung, Abschattungen, etc.) auftreten. In jedem Falle führt jedoch die bislang verwendete Modellierung (*Cell Dwell Time*) bei Wireless LAN auf eine deutliche Überschätzung der Wahrscheinlichkeit von Handoverabbrüchen.

In Abb. 3.17 sind die entsprechenden Graphen abgedruckt. Die Dauern der Handoverprozedur des Mobile IP sind im Bereich von etwa 2 s bis 4 s fast gleichverteilt, daher ist die inverse Verteilungsfunktion zunächst konstant 1 und fällt dann ab 2 s ungefähr linear bis auf Null bei etwa 4 s ab. In diesem Bereich beschreibt die bisher zur Modellierung verwendete Zellaufenthaltsdauer nahezu eine Gleichverteilung. Im Gegensatz dazu weist Wahrscheinlichkeitsdichtefunktion der hier vorgeschlagenen Zwischenhandoverzeit bei den hier gewählten Geschwindigkeiten keine Werte kleiner 4,5 s auf.

3.5 Zusammenfassung

Die in diesem Kapitel hergeleitete Verteilung der Zwischenhandoverzeiten berücksichtigt im Gegensatz zur bislang verwendeten Zellaufenthaltsdauer (*Cell Dwell Time*) auch die aufgrund der Kanalselektionsstrategie des Wireless LAN auftretende Hysterese. Diese ist einerseits durch die Empfangspegel und deren Schwellwerte, andererseits durch die geometrischen Gegebenheiten bedingt, nämlich die Position der Basisstationen und den Weg des Teilnehmers. Diese Hysterese legt eine Mindestdistanz zwischen dem Ort des Assoziierens mit der Basisstation und dem Ort des Abbruchs der Funkverbindung fest, woraus sich eine Mindestdauer zwischen zwei Handovern ergibt. Berücksichtigt wird dabei auch, daß schnellere Teilnehmer eine relativ gesehen

höhere Handoverwahrscheinlichkeit haben, so daß sich bei der hier angenommenen Gleichverteilung der Geschwindigkeiten eine Verzerrung der Wahrscheinlichkeitsdichtefunktion der Zwischenhandoverzeiten ergibt.

Für die Modellierung wurden homogene Funkwellenausbreitungsbedingungen in einer idealisierten Ebene angenommen. Durch die in realen Systemen auftretenden Effekte, wie beispielsweise Abschattungen und Interferenzen, können ebenfalls Handover ausgelöst werden. Die dann auftretende Verkürzung der Zeit zwischen Handovern wirkt sich allerdings in gleicher Weise sowohl auf die hier vorgeschlagenen Zwischenhandoverzeiten als auch auf die in der Literatur verwendete Zellaufenthaltsdauer aus.

Die hier gewonnene präzisierte Abstraktion des Handovervorgangs wird später dazu verwendet, Strategien für das Handoververhalten von mobilen programmierbaren Diensten zu evaluieren. Diese bieten umfangreiche Möglichkeiten, um die Mobilität des Teilnehmers spezifisch zu unterstützen. Dies wird ermöglicht durch die Flexibilität, die Aktive und Programmierbare Netze bieten.

Kapitel 4

Aktive und Programmierbare Netze

Verteilte Kommunikationssysteme bestehen im allgemeinen aus endlichen Zustandsautomaten, die untereinander einem gemeinsamen Protokoll folgende Nachrichten austauschen. In diesem Kontext läßt sich jeder dieser Zustandsautomaten trennen in die Bestandteile *Logik* (Beschreibung aller zulässigen Zustände und Zustandsübergänge) und *Zustand* (Parameter, Variablen, Pufferspeicher). Jeder Knoten verfügt entsprechend dem ISO/OSI-Schichtenmodell [Int93b] über einen oder mehrere Zustandsautomaten in jeder von ihm bedienten Schicht. Eine wesentliche Eigenschaft eines herkömmlichen IP-Netzes ist die auch schichtenmäßige Aufteilung der Funktionalität zwischen Endpunkten und Routern, wobei letztere durch die Zustandsautomaten der Vermittlungsschicht und erstere durch jene der Transport- und Anwendungsschicht charakterisiert sind. Daneben ist das heutige Internet aber auch dadurch gekennzeichnet, daß ein Nutzer sein Endgerät frei programmieren kann, während hingegen kaum Einflußmöglichkeiten auf die Funktionen der Router bestehen.

Bei *Aktiven und Programmierbaren Netzen* wird genau dieses ermöglicht, indem die Router dynamisch im laufenden Betrieb um neue Funktionen erweitert werden können. Erreicht wird dabei eine Öffnung des Netzes für neue Architekturen, Protokolle und Dienste, die an bestimmten Netzknoten eigene Funktionalitäten benötigen [CDK+99, CKV01]. Auch lassen sich auf diese Weise allgemein neue Leistungsmerkmale auf den Netzknoten schneller bereitstellen. Das Verhalten der im Netz bereitgestellten Protokollfunktionen kann somit mittels *Compositional Adaptation* [MSKC04] weitaus flexibler beeinflußt werden, als dies durch herkömmliche Parameteradaption möglich ist.

In [Pso99] wird ein *aktives* im Gegensatz zu einem *passiven* Netz so definiert, daß auf den dazwischenliegenden Routern eine Paketbearbeitung bis hin zu Aspekten der Anwendungsschicht erfolgen kann, die sogar vom Benutzer selbst vorgeben werden können. In [CBZS98] wird diese Eigenschaft eines aktiven Netzes als Programmierbarkeit des Paketverarbeitungsmechanismus (*Network Application Programming Interface*) definiert. In [TSS+97] werden zwei konkrete Ausprägungen vorgestellt, nämlich der *programmierbare Switch* und der *Kapsel*-Ansatz. Als *Datenkapsel* wird dabei ein IP-Paket bezeichnet, welches mit entsprechendem eingebetteten Code oder mit einer Referenz auf aus einem Repositorium zu ladenden Code versehen ist, der dann auf

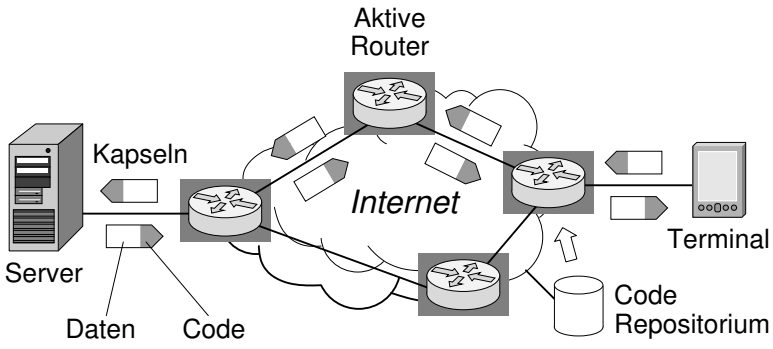


Abbildung 4.1: Aktives Netz nach dem Kapsel-Prinzip.

den aktiven Knoten ausgeführt wird. Beides ist in Abb. 4.1 schematisch dargestellt. Im Gegensatz dazu gestattet es der programmierbare Switch, Programme getrennt von den Anwendungsdaten über eine eigene Signalisierungsschnittstelle dynamisch zu laden und auf Nutzdatenströme anzuwenden.

Darauf basierend wird in der vorliegenden Arbeit nun ein System nach dem Kapsel-Prinzip als *Aktives Netz* und ein Netz aus programmierbaren Netzelementen als *Programmierbares Netz* bezeichnet. Bei diesem wird der auszuführende Code dynamisch, beispielsweise aus einem Repositorium, auf den programmierbaren Knoten geladen und dann auf alle Pakete angewandt, die einem bestimmten Kriterium entsprechen. Wie in Abb. 4.2 schematisch gezeigt, kann dies beispielsweise ein regulärer – nicht weiter für die Programmierbarkeit angepaßter – Datenfluß eines Teilnehmers zu einem Server sein. In Tabelle 4.1 werden die Eigenschaften eines Aktiven Netzes qualitativ mit denen eines Programmierbaren Netzes verglichen.

Der Unterschied zwischen Aktiven Netzen und solchen, die dem programmierbaren Ansatz folgen, tritt offen zu Tage beim Aspekt der Granularität der Steuerungsmöglichkeiten. Bei Aktiven Netzen führt jedes Paket den jeweils auszuführenden Code – oder eine Referenz – direkt mit. Bei Programmierbaren Netzen hingegen wird zuerst der erforderliche Code am passenden Ort initialisiert und dabei zugleich festgelegt, auf welche Pakete der Dienst angewandt werden soll.

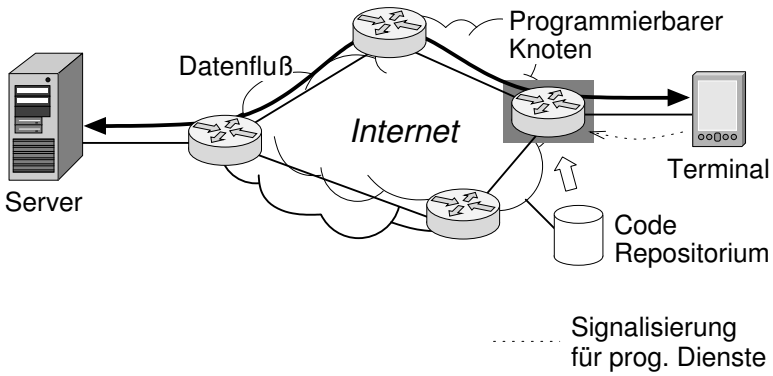


Abbildung 4.2: Programmierbares Netz mit Signalisierung durch das Terminal.

Tabelle 4.1: Vergleich der Eigenschaften von Aktiven Netzen und Programmierbaren Netzen.

Kriterium	**Aktives Netz**	**Programmierbares Netz**
Dienststeuerung	*Im Terminal*	*Terminal oder Managementinstanz*
Code-Transport	*Ganz oder teilweise mit jedem Paket*	*Einmalig je Transaktion*
Mobilität	*Inhärent unterstützt*	*Erfordert weitreichende Maßnahmen*

Erfordert nun ein Dienst unterschiedliche Bearbeitung von Paket zu Paket, ist der Overhead bei Verwendung eines Aktiven Netzes geringer. Ist hingegen die Bearbeitung für alle Datagramme eines Datenstromes gleich, so ist ein Programmierbares Netz vorteilhafter.

4.1 Netzseitiger Code, seine Verteilung und Sicherheit

4.1.1 Kapsel-basierter offener Ansatz

Das *Active Node Transfer System* (ANTS) [WGT98, WLG98] verwendet Datenkapseln anstatt regulärer IP-Pakete. Diese tragen in ihrem Header zusätzlich einen Verweis auf die passende Bearbeitungsroutine. Jeder empfangende aktive Knoten wertet die Protokollkennung aus und lädt bei Bedarf entsprechende Teile der Protokollimplementierung (Codegruppen) dynamisch nach. Unterscheiden lassen sich dabei wohlbekannte Funktionen, die auf jedem Knoten fest installiert sind, häufig verwandte Protokollfunktionen die mit hoher Wahrscheinlichkeit bereits vorhanden sind und seltener benötigte – insbesondere anwendungsspezifische – Funktionen, die dann beim Eintreffen der ersten entsprechenden Kapsel geladen werden. Sofern der absendende Knoten nicht sicher ist, ob der zur Bearbeitung seiner Datenkapseln erforderliche Programmcode im Netz vorhanden ist, wird er ihn dem ersten aktiven Router übermitteln.

Neben der Protokollkennung kann der Header der Datenkapseln noch weitere Parameter für die Bearbeitungsroutine beinhalten. In Abb. 4.3 ist die Struktur einer Datenkapsel gezeigt sowie die Anforderung und Übertragung der zur Bearbeitung notwendigen Codegruppe vom vorherigen aktiven Knoten. Dabei kann ein aktiver Knoten selbst anhand der verfügbaren Ressourcen entscheiden, ob dort eine bestimmte Protokollfunktion ausgeführt wird oder das Datagramm lediglich weitergeleitet wird.

Die Bearbeitung auf dem aktiven Knoten erfolgt zustandsbehaftet, das heißt daß eine Datenkapsel die Art und Weise beeinflussen kann, wie nachfolgende Kapseln behandelt werden. Auch können aktive Knoten untereinander Nachrichten austauschen und so die Datenübertragung über

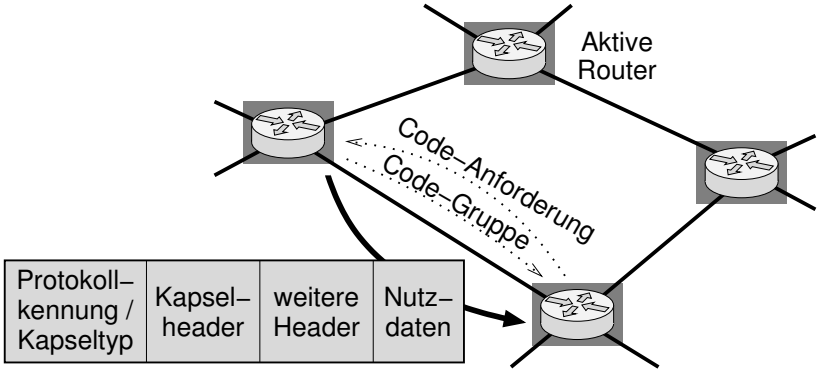


Abbildung 4.3: Daten-*Kapsel* und Codeverteilung bei ANTS.

Knoten hinweg modifizieren. Die Protokolle sind dabei in zweierlei Hinsicht strikt voneinander getrennt. Einerseits können die Zustände eines bestimmten Protokolles nicht durch Kapseln mit einer anderen Protokollkennung verändert werden. Andererseits hat die Protokollkennung bei allen Bearbeitungsschritten unverändert zu bleiben.

Die jeweilige Ausführungsumgebung sorgt auch dafür, daß Datenkapseln eines Teilnehmers nicht die Kommunikation anderer beeinflussen können. Einer indirekten Beeinflussung – beispielsweise durch speicherintensive Berechnungen – wird begegnet, indem jede Kapsel – analog zum *Time To Live* (TTL)-Feld im IP-Header – einen Ressourcenzähler beinhaltet, der entsprechend dem Ressourcenverbrauch über der Zeit dekrementiert wird. Bei der Erzeugung neuer Kapseln wird dabei der Ressourcenzähler der eingehenden Kapsel auf alle ausgehenden aufgeteilt. Der Startwert des Ressourcenzählers beschränkt also den durch die betreffende Kapsel im Netz effektiv verursachten Ressourcenverbrauch.

4.1.2 Spezialisierte Programmiersprache für aktiven Code

Die Funktionsvielfalt der gängigen Programmiersprachen könnte die Betriebssicherheit aktiver Netzen gefährden, da nachlässige – oder gar böswillige – Nutzer beispielsweise mit wenigen Anweisungen in einer Endlosschleife viele Ressourcen des Netzes belegen könnten. Es liegt jedoch in der Natur der Sache, daß die Gutartigkeit eines Programmes nicht allgemein verifiziert werden kann. In *Switchware* [AAH+98] wird daher die Verwendung einer eigenen Programmiersprache vorgeschlagen, der *Programming Language for Active Networks* (PLAN) [HKM+98]. Diese setzt auf einem zweistufigen Architekturkonzept auf.

Aktive Pakete (äquivalent zu den Datenkapseln beim oben vorgestellten ANTS) beinhalten ausführbaren PLAN-Code. Da jedoch PLAN in seiner Funktionalität hinreichend eingeschränkt ist, erübrigt sich eine Überprüfung der Funktionalität des Codes. Die aktiven Pakete beinhalten neben dem auszuführenden PLAN Code die Absenderadresse sowie die Adresse des jeweils nächsten aktiven Routers, aber nicht die des tatsächlichen Empfängers. Letztere wird erst im Laufe der Bearbeitung auf den aktiven Routern eingesetzt. Es findet also ein quellengesteuertes Routing innerhalb des aktiven Overlaynetzes, wie beispielsweise dem PLANet [HMA+99], statt.

Sogenannte *aktive Erweiterungen* gestatten es, mächtigere Funktionen zu realisieren. Diese können in universellen Programmiersprachen erstellt[1] und bei Bedarf dynamisch geladen werden. Sie stellen somit keinen mobilen Code im eigentlichen Sinne dar, sondern einen Teil der programmierbaren *Middleware*. Da die Anzahl aktiver Erweiterungen deutlich kleiner ist als die Zahl der aktiven Pakete, können hier aufwendigere Sicherheitsüberprüfungen eingesetzt werden, als sie für den in den aktiven Paketen enthaltenen Code praktikabel wären.

Das *Resource Controlled Active Network Environment* (RCANE) [AMK+01] sorgt dabei für die erforderliche Sicherheit. RCANE schottet die einzelnen Anwendungen hinsichtlich der Verwendung der Ressourcen voneinander ab und sorgt dabei auch dafür, daß jede Anwendung die erforderlichen Ressourcen wie angefordert verwenden kann. Kennzeichnend für RCANE ist eine dreistufige Unterscheidung des ausgeführten Codes. Die Basis stellt eine virtuelle Maschine dar, die in einer nativen Programmiersprache geschrieben ist. Darauf baut die mittlere Schicht auf, die aus vertrauenswürdigem Code besteht und höherwertige Funktionen zur Verfügung stellt. Indem der vom Teilnehmer übergebene Code nur auf diese, nicht aber auf die Funktionen der untersten Schicht zugreifen darf, kann die Integrität des Gesamtsystems gewahrt werden.

4.1.3 Dynamisch geladener verifizierter Code

Beim *Active Multicasting Network* (AMnet) [FHSZ02] erfolgt die Signalisierung aktiver Dienste hingegen außerhalb des Nutzdatenstromes. Dabei werden zweierlei Arten von Diensten unterstützt. Anwendungsbezogene Dienste, die typischerweise von Endsystemen initiiert werden, und von Administratoren veranlaßte netz-zentrische Dienste.

Aktive Dienste können aus einem oder mehreren dynamisch ladbaren Modulen bestehen, denen dann die entsprechenden Datenströme übergeben werden. Die Sicherheit des Codes wird dabei in einem zweistufigen Konzept realisiert. Einerseits können Dienstmodule nur aus überprüften und legitimierten Code-Repositorien mittels gesicherter Übertragung geladen werden. Alle in diesen Repositorien verfügbaren Module wurden vorab auf ihre Funktion hin überprüft. Andererseits wird der ausgeführte Code während der Laufzeit auf notorisch bekannte unzulässige Operationen hin überwacht.

4.1.4 Vergleich und Bewertung der Systeme

In Tab. 4.2 werden die drei oben dargelegten Systeme anhand der aus Netz- und Dienstarchitektursicht wesentlichen Eigenschaften verglichen. An erster Stelle steht die Frage, woher der auszuführende Code stammt. Bei ANTS wird dieser vom Endgerät falls nicht in den Datenpaketen selbst, so in einem assoziierten Signalisierungsvorgang übermittelt. Bei Switchware wird nur ein abstrahierter Code (PLAN) in den Datagrammen mitgeführt, die eigentlichen Funktionen stammen aus einem sicheren Repositorium. Bei AMnet hingegen wird ausschließlich Code aus vertrauenswürdigen Quellen geladen.

[1] Die Autoren erwähnen in diesem Zusammenhang Java und Caml.

Tabelle 4.2: Vergleich der aktiven und programmierbaren Systeme.

System	Code im Paket	Code aus Repositorium	Parameter im Paket	Konfig. *out-of-band*	Dienstinitiator	Änderung der Anwendung
ANTS	✓	–	✓	–	Endgerät	✓
Switchware	✓	✓	✓	–	Endgerät	✓
AMnet	–	✓	–	✓	Endgerät oder Netzmanagement	–

Parameter für die auszuführenden Dienste werden bei ANTS und Switchware in den Paketheadern mitgeführt, während hingegen bei AMnet diese über eine zusätzliche Signalisierungsschnittstelle übermittelt werden, so wie auch die Auswahl der auszuführenden Funktionen.

Daraus resultiert, daß Dienste bei ANTS und Switchware ausschließlich von Seiten der Endgeräte – beziehungsweise anderen auf dem Datenpfad liegenden Entitäten – initiiert werden können. Bei AMnet hingegen kann jede autorisierte Einheit einen Dienst auf einem programmierbaren Router starten und konfigurieren. Da ANTS und Switchware in die Datenpakete Code einfügen, bedingen beide Ansätze eine sende- und empfangsseitige Änderung derjenigen Anwendungsprogramme, auf deren Datenströme der Code Anwendung finden soll. Lediglich beim vollständigen *out-of-band*-Ansatz des AMnet lassen sich Dienste realisieren, die keine Modifikation von Anwendungen erfordern und für diese auch vollständig transparent sein können.

4.2 Architekturen aktiver und programmierbarer Systeme

Active Network Node (ANN) [DDPP00] Sogenannte *Router Plugins* gestatten beim ANN eine dynamische Erweiterung der Funktionalität des Routers. Der Performanz halber sind diese Plugins als Kernelmodule ausgeführt, die in den Betriebssystemkern geladen werden. Die zugrundeliegende Softwarearchitektur gestattet es, diese Plugins während der Laufzeit sowohl zu laden als auch wieder zu entfernen. Sie werden dabei von einer Steuerungseinheit in die Paketverarbeitung des Kernels eingebunden, wofür abstrahierte Schnittstellen bereitgestellt werden. Dabei können auch mehrere Instanzen des gleichen Plugins geladen werden, die gleichartige Aufgaben unabhängig voneinander bearbeiten. Schließlich stellt die Architektur einen Mechanismus zur Verfügung, der die Pakete filtert und einzelnen Strömen zuordnet, die dann wiederum bestimmten Instanzen von Plugins zugeführt werden.

Um diesen Vorgang besonders effizient zu gestalten, wird dieser Filter als *Gerichteter Azyklischer Graph* (Directed Acyclic Graph, DAG) implementiert. Hierbei werden die Regeln

entsprechend ihrem Präzisionsgrad vorgruppiert und -sortiert, so daß zuerst die allgemeineren und erst danach gegebenenfalls die spezifischeren Regeln geprüft werden. Durch die Sortierung in Form eines gerichteten Graphen läßt sich die unnötige Überprüfung von spezifischeren Regeln vermeiden, wenn bereits ein übergeordnetes allgemeineres Muster nicht zutreffend war.

Click Modular Router [KMC+00] setzen ihre Funktionen aus einzelnen atomaren Elementen zusammen. Diese sind in C++ geschrieben und werden dann mittels einer eigenen Definitionssprache deklariert und verbunden. Daraus ergibt sich schließlich die jeweilige Bearbeitungskette für bestimmte Datagramme. Die Ein- und Ausgänge der Elemente können dabei vom Typ *push* oder *pull* sein. Im ersten Fall werden Datagramme nach der Bearbeitung sofort weitergereicht (ein Beispiel wäre die Eingangsbearbeitung an einer Netzschnittstelle). Im zweiten Falle werden Datagramme beim vorhergehenden Element geholt, sobald das betreffende Element zur Bearbeitung bereit ist (wie bei der Aussendung gemäß einem Scheduling-Algorithmus). Ein- und Ausgänge unterschiedlichen Typs lassen sich über Pufferelemente koppeln. Das System gestattet eine Migration zu einer neuen Routerkonfiguration ohne den laufenden Betrieb zu unterbrechen, so daß sich der *Click Modular Router* für Aktive und Programmierbare Netze nutzen läßt.

Virtuelle Router Architektur (VERA) [KP01a] Eine verteilte virtuelle Plattform zur effizienten Ausführung von Weiterleitungsroutinen wird von der VERA bereitgestellt. Dazu wird eine Hardwareabstrahierung eingeführt, die Prozessorgruppen zu virtuellen Prozessoren zusammenfaßt, Details von Netzschnittstellen verbirgt und verteilte Warteschlangen über Prozessoren hinweg definiert.
Darauf aufbauend erstellt ein Betriebssystem eine logische Prozessorhierarchie. Innerhalb dieser Hierarchie lassen sich dann die stufenweise Paketklassifizierung, die Weiterleitungsroutinen und der Scheduler abbilden. Da diese Routerabstraktion alle für ein routendes System wesentlichen Eigenschaften einbezieht, lassen sich beispielsweise die oben genannten *Router Plugins* und der *Click Modular Router* damit konzeptionell in Kongruenz bringen.

HArPooN [**BTS+03**] unterscheidet lediglich zwischen Paketklassifizierung und -aussendung auf der einen und den Weiterleitungsroutinen auf der anderen Seite. Im Vergleich zu VERA ist dadurch keine komplexe Hardwareabstrahierung erforderlich, vielmehr werden hierbei die Weiterleitungsroutinen entsprechend ihrem Verarbeitungsaufwand auf einen Satz von programmierbaren Rechnern ausgelagert.

Bowman [MBZC00] stellt ein Betriebssystem für aktive Router dar, welches als Basis für unterschiedliche Ausführungsumgebungen dienen kann, wie beispielsweise auch für ANTS. Bowman beinhaltet drei Abstraktionen, auf denen aktive Dienste aufsetzen können. Einerseits können von aktiven Diensten *Kanäle* definiert werden, die dann von den Diensten als Kommunikationsendpunkte genutzt werden können.
Der Bearbeitungsfluß innerhalb eines aktiven Routers einschließlich der Verarbeitungszustände wird hier als *A-Flow* bezeichnet. Schließlich gibt es einen *Zustandsspeicher*, in

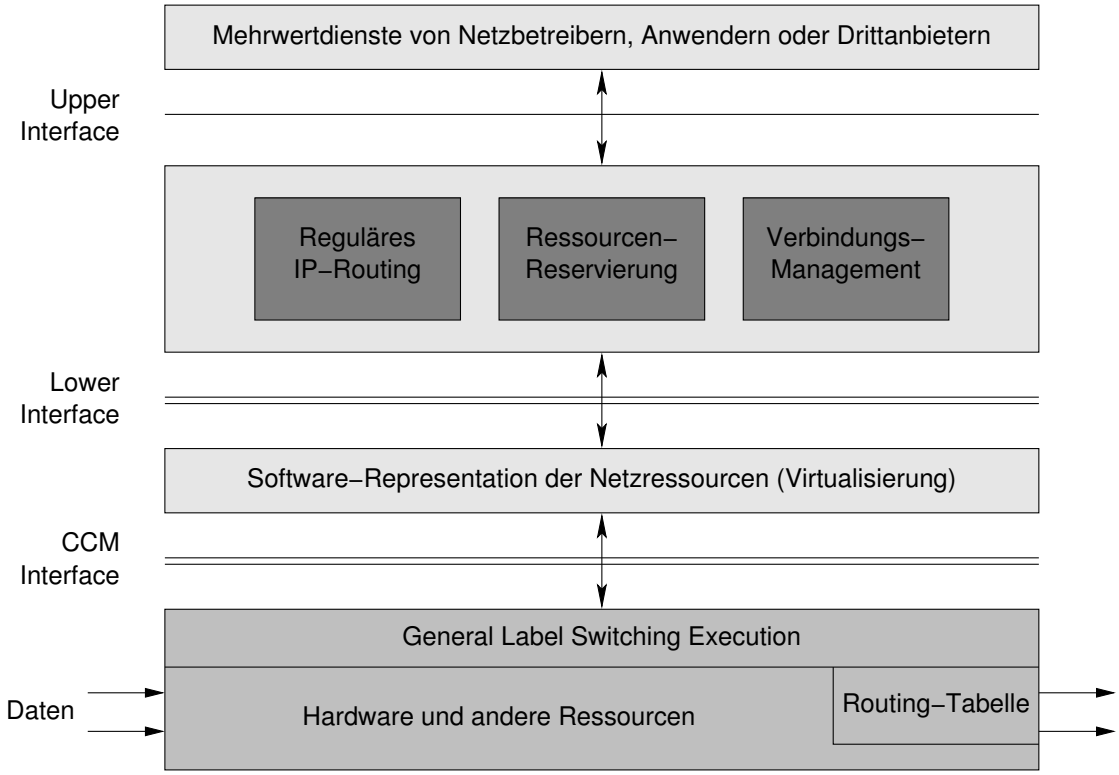


Abbildung 4.4: Das P1520-Referenzmodell für IP-Router und Switche.

dem *A-Flows* Werte ablegen und auch übergeben können, ohne daß es dazu einer direkten Kopplung von Prozessen und -variablen bedürfte.

Lara++ [SFSS01] Die Erweiterung der *Lancaster Active Router Architecture* (LARA) zieht im Gegensatz zu den bisher genannten Architekturen auch in Betracht, daß Dienste eines Teilnehmers einander nicht notwendigerweise vertrauen und auch hier eine Abschottung erforderlich sein kann. Dazu werden jeweils diejenigen Dienstmodule eines Teilnehmers, die einander explizit vertrauen, in einer eigenen Prozeßumgebung innerhalb der Ausführungsumgebung gestartet.

Referenzmodell P1520 [BLH+98] Das *P1520*-Referenzmodell ist ein Architekturkonzept für programmierbare Netzknoten, welches sich in seiner Schichtung stark an den zu erbringenden Telekommunikationsdiensten orientiert. Es ist der einzige übergreifende Entwurf, der gleichermaßen für Ethernet, IP, ATM und SS7 gedacht war. In Abb. 4.4 ist ein Überblick über das für IP-Router und Switche in [LDV+99] definierte Referenzmodell P1520 zu finden. Ein erster Abstraktionsschritt erfolgt durch die als *Connection Control and Management* (CCM) bezeichnete Schnittstelle. Diese gestattet den Zugriff auf die unterste Ebene der IP-Funktionen und sollte zugleich Trigger im Falle des Eintreffens bestimmter Pakete unterstützen. Das *Lower Interface* verschafft Zugriff auf die Steuerungs- und Management-Elemente für das IP-Routing. Darauf bauen die Funktionen für Routing, Verbindungsma-

nagement, Ressourcenreservierung, etc. auf. Diese wiederum bieten ihre Funktionalität über das *Upper Interface* den darauf aufsetzenden Mehrwertdiensten an, wie beispielsweise Virtuelle Private Netze. Diese können von den Netzbetreibern, den Nutzern selbst oder auch Drittanbietern erstellt sein.

Die wesentlichen Eigenschaften aller vorgestellten Architekturen sind einerseits der *strukturierte Zugriff* auf Datagramme und die Bearbeitung selbiger durch einen oder mehrere Funktionsroutinen, die in ihrer Summe höherwertige Dienste (einschließlich weiterer Signalisierung) realisieren. In Abschnitt 4.7 wird daraus ein idealisiertes Modell einer programmierbaren Plattform abgeleitet.

4.3 Dienste in Aktiven und Programmierbaren Netzen

4.3.1 Designkriterien für Dienste

Basierend auf der *Middleware* der aktiven und programmierbaren Router lassen sich nun spezifische Dienste realisieren. Aus Systemsicht sind für das Erbringen von Ende-zu-Ende-Diensten dreierlei Punkte entscheidend.

Dienststeuerung Es sind zwei Aspekte der Dienststeuerung zu unterscheiden, die Initiierung des geeigneten Dienstes am geeigneten Ort einerseits und andererseits dessen Konfiguration. Bei Aktiven Netzen beinhaltet jede Kapsel eine Referenz auf den darauf anzuwendenden Dienst. Daher kann ein Dienst auch nur von der Terminalseite her initiiert werden. Allerdings können die Dienste gegebenenfalls auch Signalisierungsnachrichten von anderen Entitäten als dem jeweiligen Terminal empfangen, die die Zustände des Dienstes modifizieren. Bei Programmierbaren Netzen hingegen verfügt die jeweilige programmierbare Plattform über einen separaten Signalisierungskanal. Über diesen können Dienste auch von Dritten, beispielsweise einer Administrationsinstanz, initiiert werden. Auch können Dienste auf beliebige, nicht besonders gekennzeichnete Pakete angewandt werden. Da Pakete hierbei einen Dienst nicht einfach umgehen können, sind Programmierbare Netze auch für Zwecke der Netzadministration geeignet.

Kompatibilität mit bestehenden Anwendungen Die Verwendung zusammen mit bereits existierenden Systemen, Protokollen und Anwendungen ist ein wesentlicher Aspekt netzseitiger Dienste. Prinzipiell sind bei Aktiven Netzen bedingt durch die damit verbundene *in-band* Signalisierung die entsprechenden Anwendungsprogramme so zu modifizieren, daß sie die Fähigkeiten des Netzes nutzen und den passenden Code einschließlich Parametern in die Pakete einfügen können. Bei Programmierbaren Netzen erfolgt die Signalisierung hingegen *out-of-band*, so daß sie auf einfache Weise auch von einem anderen Programm als der Anwendung selbst durchgeführt werden kann.

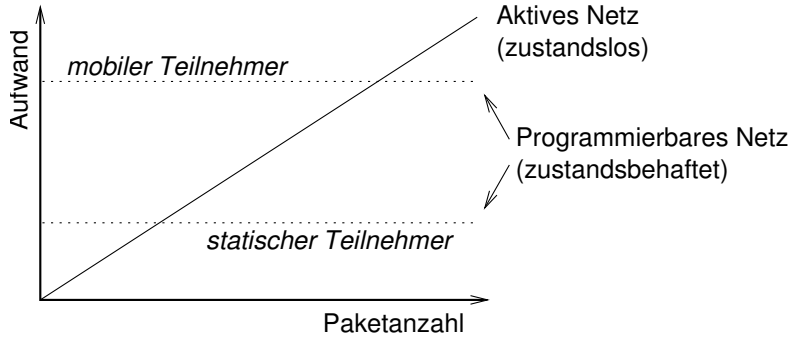


Abbildung 4.5: Aufwand für die Dienststeuerung im schematischen Vergleich.

Mobilität Wechselt ein Endgerät seinen Netzzugangspunkt, kann eine Rekonfiguration der genutzten Dienste erforderlich sein. Bei einer zustandsbehafteten Verarbeitung im Netz, wie sie für Programmierbare Netze typische ist, macht dies einen Transfer des entsprechenden Kontextes erforderlich. Bei Aktiven Netzen hingegen kann die Zustandshaltung auf das Endgerät beschränkt werden, was sich gerade im Falle häufiger Netzwechsel als vorteilhaft erweisen kann. In Abb. 4.5 ist der Aufwand für die Dienststeuerung je Datenfluß schematisch dargestellt. Bei zustandslosen Systemen steigt dieser proportional zur Paketanzahl, während er bei zustandsbehafteten Systemen davon unabhängig ist. Im Gegenzug erhöht die Mobilität eines Teilnehmers den Aufwand nur im zustandsbehafteten Fall.

4.3.2 Nomenklatur der Dienste

Für einen derart zentralen Begriff wie den des *Dienstes* gibt es im Bereich der Informations- und Kommunikationstechnik verschiedene Definitionen. Bei der in der Tradition der Telegraphie und Telephonie stehenden *International Telecommunication Union* (ITU) ist ein Dienst dasjenige, "was ein Betreiber seinen Kunden anbietet um ihnen damit einen bestimmten Telekommunikationsbedarf zu erfüllen" [Int93a]. Beispiele sind die Dienste Fernsprechen, Telefax, oder Datenübertragung. Diese zunächst sehr allgemeinen Beschreibungen werden durch eine Vielzahl von Anschluß- und Basisdienstmerkmalen konkretisiert – z. B. die Art der Teilnehmeranschlußeinrichtung, des Auf- und Abbaus von Verbindungen – sowie durch ergänzende Dienstmerkmale, wie beispielsweise Anklopfen, geschlossene Benutzergruppen, etc. [Boc90]. Im Gegensatz dazu steht der Begriff des Dienstes in der sich eher an der Datenverarbeitung orientierenden *Internet Engineering Task Force* (IETF) für "einen Prozeß oder ein System, welches dem Netz einen bestimmten Nutzen erbringt" [VGPK97].

Diesen beiden Definitionen liegen unterschiedliche Auffassungen der im Netz ablaufenden Vorgänge zu Grunde. Auf der einen Seite steht die netzorientierte, "klassische" Telekommunikationstechnik (repräsentiert durch die ITU), die einen Dienst als etwas betrachtet, was nach einer Anforderung durch den Teilnehmer vom Netz erbracht und auch vollständig von den Netzeinheiten gesteuert wird. Dem gegenüber steht die endgeräteorientierte Einstellung der IETF, bei der

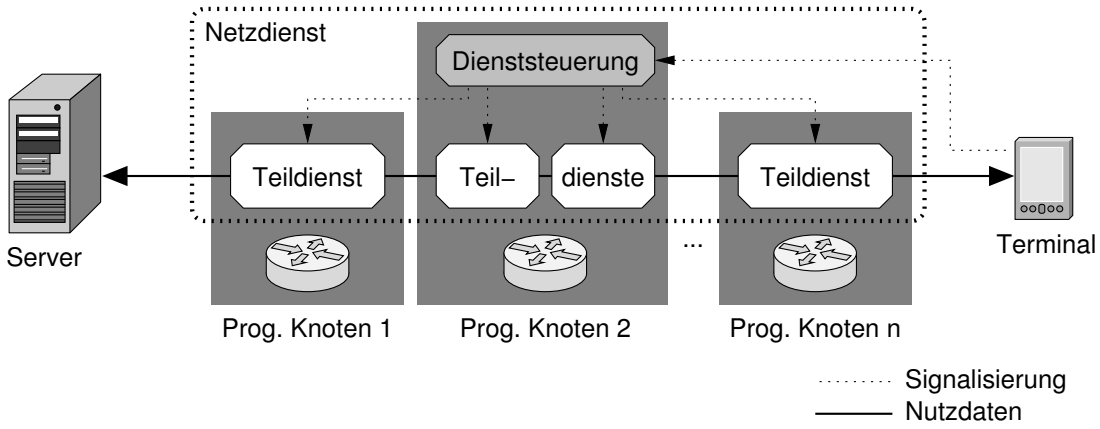


Abbildung 4.6: Netzdienst bestehend aus Teildiensten und zentraler Dienststeuerung.

ein Teilnehmer für einen bestimmten Kommunikationsvorgang alle erforderlichen Dienste bei entsprechenden Servern anfordert und gegebenenfalls auch selbst steuert.

Um die Begrifflichkeiten klar trennen zu können, wird im Rahmen dieser Arbeit für den Bereich der Aktiven und Programmierbaren Netze in Anlehnung an [CBZS98, CDK+99] der Begriff des *Netzdienstes* verwendet:

> *Ein* Netzdienst *ist die Gesamtheit aller Maßnahmen, die das zielgerichtete Bearbeiten der von einem Endgerät übermittelten Daten in einem oder mehreren aktiven beziehungsweise programmierbaren Knoten über das reguläre IP-Routing hinaus ermöglichen.*

Im einfachsten Fall besteht ein derartiger Netzdienst aus einer einzigen Entität, die vom Endgerät direkt an einem bestimmten Ort instantiiert und initialisiert wird. Im Verlauf der Nutzung dieses Netzdienstes muß das Terminal selbst alle erforderlichen Maßnahmen treffen, um ihn stets den aktuellen Bedürfnissen anzupassen. Das entspricht der oben erwähnten endgeräte-zentrischen Definition der IETF.

Komplexere Netzdienste sind in der Regel aus mehreren Entitäten zusammengesetzt, die wiederum an mehreren Orten beheimatet sein können. Damit diese Teildienste zusammen den gewünschten Netzdienst erbringen, ist eine Koordinierung durch eine *Dienststeuerung* erforderlich. Diese kann sowohl im Endgerät des Teilnehmers (terminal-zentrischer Fall) als auch in einem Netzknoten (netz-zentrischer Fall) angeordnet sein. Letzteres ist in Abb. 4.6 schematisch gezeigt. Man erkennt, daß der Netzdienst dann eher die Dienstdefinition der ITU erfüllt, während hingegen der Dienstbegriff der IETF auf die hier als Teildienste bezeichneten Einheiten zutrifft. Die beiden Definitionen des Dienstbegriffs der ITU und der IETF sind also nicht so disjunkt wie es zunächst erscheint, sondern beschreiben lediglich unterschiedliche Aspekte der Diensterbringung in Kommunikationsnetzen.

Die Unterscheidung zwischen Teildiensten und Dienststeuerung ist dabei nur von logischer Natur, beide sind (statische oder dynamisch ladbare) *Features* (Leistungsmerkmale) der program-

mierbaren Knoten (beziehungsweise Endgeräte). Die Leistungsmerkmale der Teildienste sind dabei für die eigentliche Bearbeitung der Pakete in den jeweiligen programmierbaren Routern – entsprechend den von der Dienststeuerung vorgegebenen Parametern – zuständig, wie im nächsten Abschnitt dargelegt.

Die Aufgaben der Dienststeuerung umfassen einerseits das Laden und Initialisieren der Dienste. Dies kann sowohl auf Anforderung eines Endanwenders als auch des Administrators hin erfolgen. Andererseits sind Ort(e) und Funktionen der Teildienste so zu wählen, daß der Netzdienst bestmöglich erbracht wird. Zu berücksichtigen sind dabei weitere Aspekte wie *Feature Interaction* (siehe Abschnitt 4.5.1) und auch die Mobilität, wozu in Abschnitt 5.2 ein neuer Ansatz vorgestellt wird.

4.3.3 Leistungsmerkmale (Features) auf IP-Ebene

Allgemein wird ein *Feature* in [CV93] definiert als "das einem bestehenden System hinzugefügte Inkrement an Funktionalität". Der tatsächliche Umfang eines einzelnen Features und seine Komplexität sind dabei einer der Freiheitsgrade beim Entwerfen von Diensten und nicht a priori festgelegt[2]. Läßt man es zu, daß ein *Feature* andere *Features* (oder rekursiv auch sich selbst) aufrufen kann, lassen sich durch derartige Kombinationen aus Leistungsmerkmalen von an sich geringer Komplexität aufwendige Dienste gestalten. Ein besonderer Vorteil der so erhaltenen Modularität ist die Möglichkeit, auf bestehende Leistungsmerkmale bei der Erstellung anderer zurückzugreifen. Die Wiederverwendung von Leistungsmerkmalen ist jedoch nur ein hinreichendes, aber kein notwendiges Merkmal modularer Systeme.

Die grundlegenden Möglichkeiten der Handhabung von IP-Paketen in einem aktiven oder programmierbaren Knoten sind in Form einer Ontologie in Abb. 4.7 schematisch dargestellt. Ausgehend von den vier Kategorien der Paketverarbeitung (*bearbeiten*, *erzeugen*, *terminieren* und *verwerfen*) werden dabei alle Varianten sowie deren notwendige und konditionale Abhängigkeiten aufgezeigt.

Die Interaktion der Bearbeitung von IP-Paketen mit den darüberliegenden höheren Schichten hat zwei Aspekte. Einerseits kann eine Bearbeitung von Feldern des IP-Headers eine Änderung der Nutzdaten bedingen, beispielsweise wird beim *File Transfer Protocol* (FTP) die IP-Adresse des Clients in den IP-Nutzdaten übermittelt [PR85] und muß folglich bei einer *Network Address and Port Translation* (NAPT) [SE01] auch dort angepaßt werden. Andererseits kann eine Bearbeitung der IP-Nutzdaten auch gemäß den Erfordernissen einer höheren Schicht gewünscht sein. In beiden Fällen muß im jeweiligen Leistungsmerkmal eine ausreichende Teilmenge der Logik der höheren Schicht implementiert sein.

[2] Tatsächlich wird der Begriff des Leistungsmerkmals für sehr unterschiedliche "Inkremente" verwandt. Beispielsweise können ergänzende Dienste (z. B. die Anrufumlenkung beim Telefondienst) auf der Basis von Leistungsmerkmalen der Vermittlungssysteme realisiert werden [Neu02].

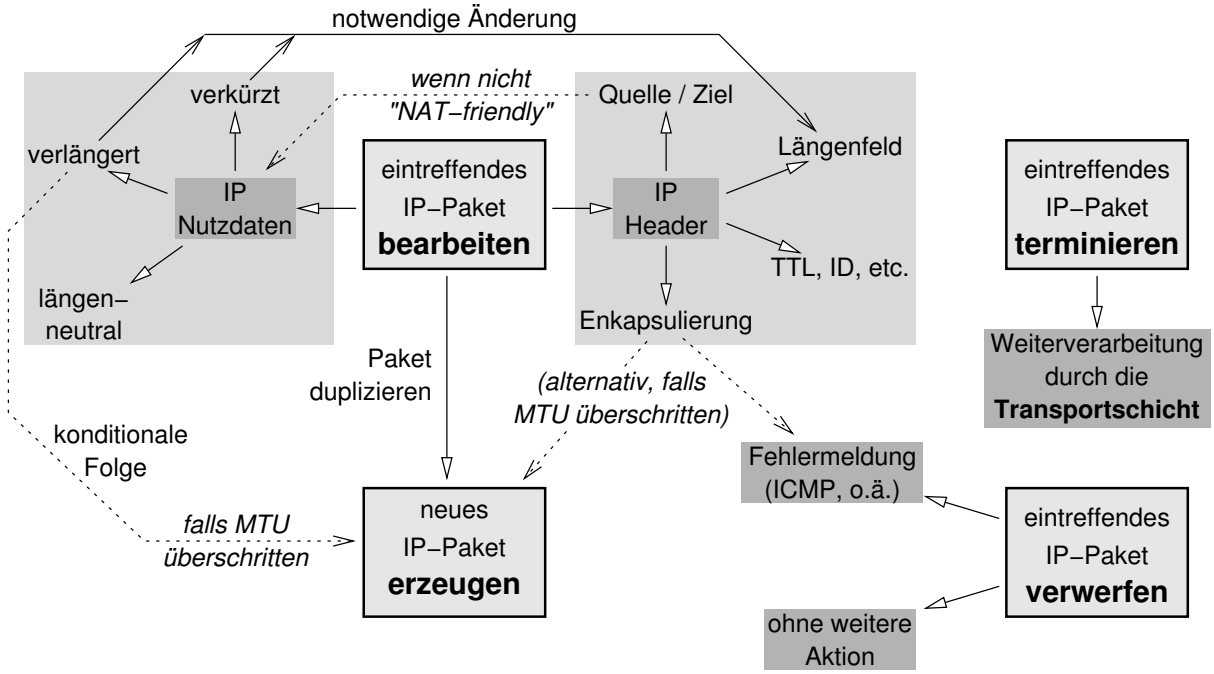


Abbildung 4.7: Ontologie der Bearbeitungsmöglichkeiten eines IP-Paketes.

4.3.4 Explizit und implizit initiierte Netzdienste

Allgemein kann ein Netzdienst *explizit* initiiert und gesteuert werden, indem ein Teilnehmer über eine Signalisierungsschnittstelle diesen anfordert. Bei Aktiven Netzen erfolgt das Signalisieren von Seiten des jeweiligen Endgerätes *in-band* innerhalb des Nutzdatenstromes. Die Signalisierungsdaten, also die entsprechenden Befehle und Parameter, müssen dabei von jeder Anwendung selbst in die Pakete eingefügt werden. Daher können nur solche Anwendungen auf die Dienste eines Aktiven Netzes zurückgreifen, die über diese besondere Funktionalität verfügen. Darüber hinaus muß die Anwendung gegebenenfalls auch über Kenntnisse des Netzzustandes verfügen, um den Dienst adäquat steuern zu können. Von Vorteil ist dabei aber, daß die Anwendung den Dienst paketweise ihren spezifischen Anforderungen anpassen kann.

Bei Programmierbaren Netzen erfolgen Initiierung und Steuerung *out-of-band* über eigene Signalisierungskanäle. Dabei wird gemäß einer Anfrage eine neue Instanz des Dienstes erstellt und entsprechend den vom Endteilnehmer übergebenen Parametern sowie den im Netz vorhandenen Kenntnissen über dessen Zustand konfiguriert. Die Signalisierung kann dabei einerseits wie bei Aktiven Netzen von der Anwendung selbst ausgehen. Andererseits kann die Steuerung von einer eigenen Anwendung oder dem Betriebssystem durchgeführt werden. Dadurch lassen sich auch solche Dienste steuern, die mehrere Anwendungen eines Terminals betreffen, wie beispielsweise Netzdienste zur Mobilitätsunterstützung.

Des Weiteren lassen sich auf diese Weise *transparente* Netzdienste realisieren, die für die initiierende und verwendende Instanz ohne Änderung des ursprünglichen Protokollverhaltens nutzbar sind. Netzdienste für administrative Zwecke werden zwar nicht vom Endnutzer selbst initiiert,

sind deshalb aber nicht zwangsläufig transparent. Eine explizite Signalisierung kann erforderlich sein, beispielsweise um Identifizierungsdaten für AAA-Dienste zu übertragen.

Ferner kann in Programmierbaren Netzen ein Netzdienst auch *implizit* initiiert werden, also dann wenn bestimmte vorab definierte Umstände eintreten (beispielsweise ein bestimmtes Protokoll verwendet wird). Allerdings läßt sich dies wiederum auf den Fall eines explizit vom Netzmanagement initiierten Netzdienst zurückführen. Dieser erkennt vorab benannte Zustände und leitet daraufhin die entsprechenden Maßnahmen zum Start eines neuen Dienstes ein.

Voraussetzung für das explizite Nutzen von programmierbaren Diensten durch das Terminal ist dabei, daß das Terminal den Ort passender Router kennt. Bei Aktiven Netzen wird es seine Kapseln dorthin adressieren, bei Programmierbaren Netzen eine entsprechende Dienstanfrage initiieren.

4.3.5 Lokalisieren von Ressourcen und Diensten (Service Discovery)

Dienste zum Auffinden von Diensten verringern den Aufwand zur Konfiguration von Geräten, was insbesondere bei der Kommunikation mobiler Terminals von besonderer Bedeutung ist [Ric00]. Dafür gibt es in klassischen IP-basierten Netzen bereits einige Lösungen. Beispielhaft seien im folgenden kurz das *Service Location Protocol* und CORBA vorgestellt.

Service Location Protocol (SLP) [VGPK97] Das SLP ermöglicht das Auffinden des Ortes (der IP-Adresse) eines Dienstes anhand der vom Nutzer gewünschten Diensteigenschaften. In einem einfachen Fall sendet der Nutzer dazu seine Anfrage an eine dienstspezifische Multicast-Adresse. Alle Dienste, die die Anforderungen erfüllen, werden daraufhin antworten. Da diese Multicast-basierte Lösung nicht beliebig skaliert, können in größeren Netzen *Directory Agents* zum Einsatz kommen, die Informationen über verfügbare Dienste sammeln und als zentrale Anlaufstelle für Anfragen der Terminals dienen. Die Verwendung des Multicast läßt sich vollständig vermeiden, indem die Adresse des Directory Agent in den DHCP-Nachrichten bekanntgegeben wird.

Common Object Request Broker Architecture (CORBA) [Vin97] bietet zwei verschiedene Dienste zur Service Discovery an. Einerseits gestattet der *Naming Service* das Auffinden von Diensten anhand des Namens, andererseits gibt es den *Trading Service,* der Objekte entsprechend den geforderten Eigenschaften lokalisiert. (Zu CORBA siehe auch Abschnitt 4.6.2)

Bei Aktiven und Programmierbaren Netzen kommt nun die Besonderheit hinzu, daß Dienste nicht fest sind, sondern im Rahmen der Möglichkeiten an die Anforderungen des Terminals dynamisch angepaßt werden können. Insofern ändert sich hier die Aufgabe der Dienstlokalisierung, da nun nicht nach einem konkreten Dienst, sondern vielmehr nach einer kompatiblen Plattform gesucht wird. Diese Plattform läßt sich jedoch wiederum als Dienst definieren. Daher können die

oben beschriebenen Mechanismen mit Multicast und/oder zentralem beziehungsweise dezentralem Verzeichnisdienst – die in dieser oder ähnlicher Weise jeder Methode zur Dienstfindung zu Grunde liegen – ebenfalls beim Auffinden einer passenden programmierbaren Plattform Verwendung finden.

4.4 Beispiele von Diensten

Analog zum *MBone* [Eri94] wurde ein *ABone* [BBR00] geschaffen. Dieses besteht aus aktiven Routern, welche mittels eines *Virtual Link Layers* scheinbar direkt – tatsächlich jedoch über IP- oder UDP-Tunnel – miteinander verbunden sind. Das *ABone* diente[3] dazu, Aktive Netze und darauf aufbauende Dienste standortübergreifend zu erproben. Nachfolgend seien exemplarisch einige nützliche aktive Dienste vorgestellt.

4.4.1 Active Reliable Multicast

Die beim Unicast zur Sicherung des Datentransports übliche Bestätigung von Datagrammen läßt sich beim Multicast nicht einsetzen, da es hier eine große Zahl von Empfängern geben kann. Daher wird bei Verfahren zur gesicherten Übertragung lediglich im Falle des Fehlens eines Datagramms eine negative Benachrichtigung (*NACK*) gesendet. Der Empfänger wertet dazu die Sequenznummern aus. Da IP-Pakete jedoch auch in ihrer Reihenfolge geändert werden können, muß noch eine gewisse Zeit abgewartet werden, bevor für eine fehlende Sequenznummer ein *NACK* gesendet wird. Der Sender wird dann das entsprechende Paket nochmals versenden. Dies führt jedoch einerseits zur sogenannten *NACK-Implosion*, falls ein Paket bei mehreren Sendern nicht ankommt. Andererseits werden Ressourcen verschwendet, da jede Wiederholungssendung an die gesamte Multicast-Gruppe gesendet wird.

Der *Active Reliable Multicast* [LGT98, SKB+01] fügt nun aktive Router in den Multicast-Baum ein. Diese speichern die Datenpakete zwischen, um *NACKs* lokal bedienen zu können. Nur wenn ein Paket nicht im Speicher ist, wird ein einziges *NACK* an den davorliegenden aktiven Router gesendet, selbst wenn mehrere Terminals ein *NACK* für das gleiche Paket senden. Die Sendewiederholung wird aber in jedem Fall nur an den Teil des Multicast-Baumes gerichtet, von dem aus ein *NACK* gesendet wurde. Auf diese Weise wird das Skalierungsproblem des gesicherten Multicast gelöst.

Die Technologie der Aktiven Netze gestattet es jedoch, darüber hinausgehende, *protokollspezifische* Funktionen einzuführen. Ist dem aktiven Router bekannt, wo sich die Sequenznummern der Anwendung in den Datagrammen befinden und wie diese zu interpretieren sind, so kann bereits der aktive Router das Fehlen eines Paketes erkennen und bereits vor den Terminals ein *NACK* in Richtung Quelle senden. Das fehlerhafte Paket wird dann aus dem Zwischenspeicher nachgereicht. In Abb. 4.8 ist einerseits dieses dargestellt, andererseits auch wie ein Terminal ein

[3] Unter anderem aus der *ABone*-Initiative heraus entstand das *PlanetLab* [CCR+03].

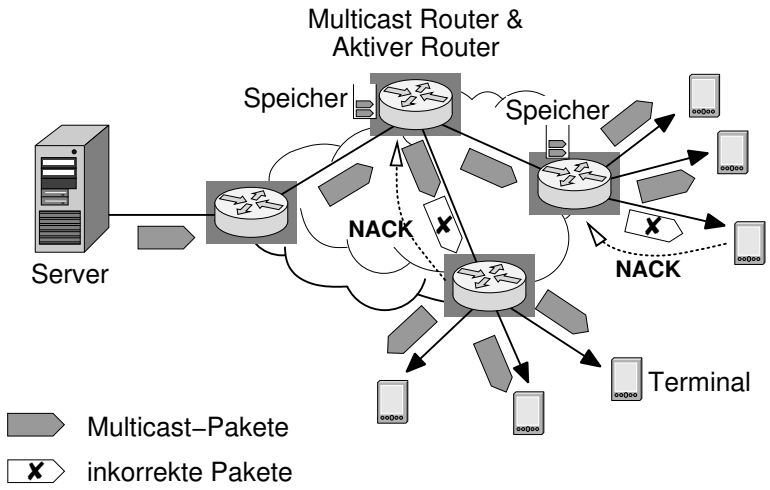


Abbildung 4.8: Verhindern der *NACK*-Implosion beim gesicherten Multicast.

NACK aussendet und anschließend vom vorangehenden aktiven Router eine gespeicherte Kopie zugestellt bekommt.

Daneben kann eine Analyse der *NACKs* ausgeführt werden, die die Paketfehlerraten (beziehungsweise Rate der *NACKs*) einzelner Teilnehmer erfaßt. Die aktiven Knoten werden dann den größten Wert ermitteln und nur diesen an den Sender weiterreichen. Auf diese Weise kann der Sender feststellen, ob die Datenrate für diese Multicast-Sitzung reduziert werden muß, um an die Gegebenheiten des "schlechtesten" Empfängers angepaßt zu sein.

Für diese Funktionen benötigen die aktiven Router nicht nur die Fähigkeit, Pakete auslesen und duplizieren zu können. Sie müssen auch in der Lage sein, höhere Zusammenhänge zwischen Paketen auswerten und angepaßte Zeitintervalle abwarten zu können, bevor eine Antwort generiert wird. Für die Anpassung der Datenrate muß sogar ein eigenes Protokoll mit all seinen Zustandsautomaten im aktiven Router ablaufen. Derlei Funktionen gehen weit über die bloße Modifikation durchlaufender Pakete hinaus und stellen komplexe Anforderungen an die Plattform.

4.4.2 Netzseitige Ratenadaptierung für Multimediaströme

Bei den heute gebräuchlichen Ansätze zur Adaptierung der Datenrate eines Multimediastromes an die aktuelle Netzlast meldet der Empfänger seine Paketfehlerrate an den Sender, der daraufhin die Senderate senkt beziehungsweise erhöht. Bei Multicast-Gruppen führt dies dazu, daß die Qualität auf den kleinsten gemeinsamen Nenner gesenkt wird. Um nun die Datenrate gezielt für einen bestimmten Empfänger zu reduzieren, wird in [KCD+00] die Kombination einer Wavelet-basierten Bildcodierung des Senders mit einem passenden Paketfilter im aktiven Router vorgeschlagen.

Das *Wavelet*-basierte Bildcodierungsverfahren verteilt die Information entsprechend ihrer Frequenzhöhe auf Datagramme, so daß hochfrequente und niederfrequente Anteile in unterschiedli-

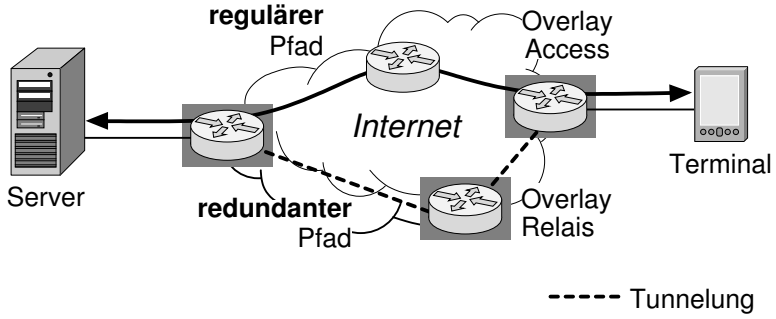


Abbildung 4.9: Architektur des PIRST-ON.

chen Datagrammen beinhaltet sind. Der aktive Router lädt nun eine entsprechende Funktionslogik, die diese Information erfassen und so die Auswirkung eines jeden Paketes auf die Bildqualität beim Empfänger bewerten kann. Genügt die verfügbare Datenrate nicht für den gesamten Datenstrom, so wird der aktive Router zunächst höherfrequentere Anteile verwerfen, wodurch lediglich die Detailschärfe des Bildes gesenkt wird, der Bildaufbau jedoch intakt bleibt.

4.4.3 Fehlertolerante Overlaynetze

Beim *Programmable, Intermediate, Resilient, Self-Configuring, Transparent Overlay Network* (PIRST-ON) [**BT02**] wird die Technologie der Programmierbaren Netze in einer für Anwendungsprogramme und -protokolle transparenten Art und Weise genutzt. Wie in Abb. 4.9 gezeigt, wird hier das Gateway des Zugangsnetzes mit der Funktionalität eines programmierbaren Knotens ausgestattet. Verfügt das Gateway des Kommunikationspartners (Servers) ebenfalls über die Möglichkeit zur Programmierbarkeit, so können für die Datenströme der Anwendungen zum Originalpfad redundante Pfade konfiguriert werden. Dazu dienen Relaispunkte (z. B. auf Gateways anderer Zugangsnetze), an die der Datenstrom in einem IP-in-IP-Tunnel gesendet wird. Diese *Overlay-Relais* dekapsulieren jedes Datagramm und senden es dann mittels regulärem Routing an den Empfänger weiter.

Die im Zugangsgateway befindliche *Overlay-Access*-Funktion analysiert dabei die Datenströme der Anwendungen hinsichtlich Paketverlusten. Im Falle der 1:1-Sicherung wird erst dann auf den redundanten Pfad umgeschaltet, wenn zu große Paketverluste auftreten. Bei 1+1-Sicherung hingegen werden per se alle Pakete dupliziert und über beide Wege übertragen. Im Zugangsgateway der empfangenden Seite werden dann die Duplikate der Datagramme verworfen, wozu die protokollspezifischen Sequenznummern ausgewertet werden. Da eine Vielzahl an Protokollen im Einsatz ist, die nicht notwendigerweise vorab bekannt sind, bietet es sich an, diese Funktionen auf dem Zugangsgateway dynamisch zu laden.

4.4.4 Programmierbarkeit auf der Anwendungsschicht

Das *Application Level Active Network* (ALAN) [FG99] erlaubt es, mit Hilfe *Dynamischer Proxy Server* bestimmte Anwendungen wie beispielsweise Abrufe von WWW-Seiten oder Multimediainhalten zu optimieren. Erkennt der *Dynamische Proxy Server*, daß für eine bestimmte Transaktion eine entsprechende Funktion (*Proxylet*) existiert, wird diese dynamisch geladen. Nützliche Funktionen sind beispielsweise die Komprimierung von Dateien vor dem Transport über kapazitätslimitierte Leitungen oder Umsetzung der Übertragung auf passendere Protokolle.

Im Vergleich mit den anderen hier vorgestellten netzschichtlastigen Diensten ist die Programmierbarkeit rein auf die Anwendungsschicht beschränkt. Somit erledigt bereits die Transportschicht die Trennung der Datenströmen einzelner Nutzer.

4.4.5 Dynamischer Einsatz von Transportprotokollen

Einen auf die Endpunkte einer Transportverbindung bezogenen programmierbaren Ansatz stellen die *Self-spreading Transport Protocols* (STP) [PWW+03] dar. Diese ermöglichen die dynamische Verbreitung von Protokollimplementierungen. Auf diese Weise kann beispielsweise ein WWW-Server allen Clients eine neue Variante des TCP oder ein mobiles Terminal jedem Kommunikationspartner eine mobilitätsunterstützende TCP-Implementierung (vgl. Abschnitt 2.4.1) zur Verwendung übergeben. Um die Kompatibilität mit nicht STP-fähigen Kommunikationspartnern zu gewährleisten, kommt ein lediglich um zusätzliche Optionen beim Verbindungsaufbau bereichertes TCP zum Einsatz. Erst wenn während des Verbindungsaufbaus festgestellt wird, daß beide Endpunkte über STP verfügen, folgen die spezifischen Nachrichten zur Aushandlung des dynamischen Transportprotokolls.

Verfügt die Gegenstelle nicht über eine entsprechende Implementierung des ausgehandelten Transportprotokolles, so wird – um zusätzliche Verzögerungen zu vermeiden – der aktuelle Datentransfer über reguläres TCP geführt. Parallel dazu wird die neue Protokollimplementierung heruntergeladen und steht so für zukünftige Datenübertragungen zur Verfügung.

Da der Code auch aus nicht vertrauenswürdigen Quellen stammen kann, sieht STP entsprechende Mechanismen zur Kontrolle seiner Funktionalität vor. Dazu wird der Quellcode der Protokollimplementierung geladen und auf dem verwendenden Rechner selbst compiliert. Dies ermöglicht einerseits den Quellcode selbst auf Konformität zu prüfen und andererseits besondere Prüffunktionen einzufügen, die eine Kontrolle des Programms während der Laufzeit gestatten. Erst dann wird das Programm in den Betriebssystemkern geladen. Die Ausführung erfolgt in einer abgeschotteten Umgebung, so daß das Programm nur Zugriff auf bestimmte, zulässige Funktionen hat. Insbesondere wird kein Zugriff auf Daten anderer Transportverbindungen oder gespeicherte Daten zugelassen.

Schlußendlich wird die Funktionalität selbst mittels einer Quellflußsteuerung überwacht. Diese muß sicherstellen, daß die Datenrate des Transportprotokolls sich kooperativ gegenüber TCP-Datenströmen verhält, also nicht mit einer höheren Datenrate sendet als dies TCP tun würde.

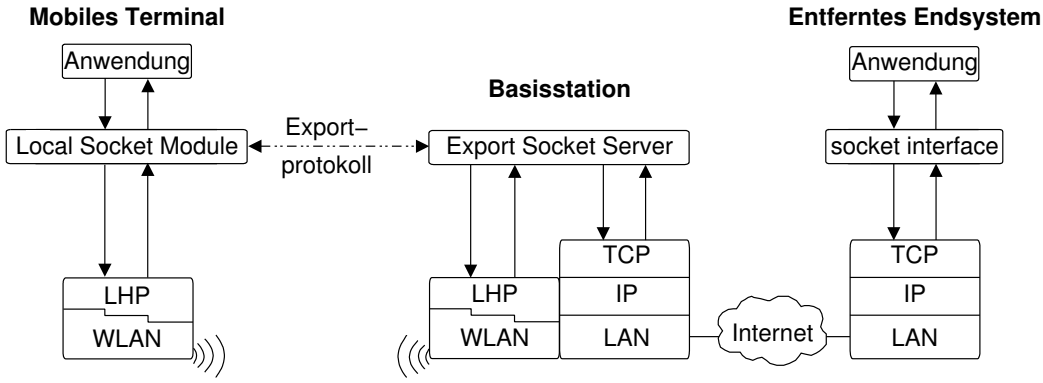


Abbildung 4.10: Überblick über die *Remote Socket Architecture* (ReSoA) nach [SRBW01].

Dazu schätzt die Quellflußsteuerung die maximale TCP-Datenrate und verwirft überzählige Pakete. Da jedoch das Paketdatenformat der Transportprotokolle beliebig gewählt sein kann, muß die entsprechende Instanz der Transportprotokollimplementierung dazu die jeweiligen Sequenznummern übermitteln. Daneben werden auch die IP-Empfänger- und Absenderadresse jedes Datagramms auf Korrektheit überprüft.

4.4.6 Remote Socket Architecture (ReSoA)

Die *Remote Socket Architecture* (ReSoA) [SRBW01, Sch04] ist eine Proxy-orientierte Architektur für den drahtlosen Internetzugang über Wireless LAN (WLAN). Im Unterschied zu herkömmlichen Systemen wird TCP jedoch nur für den Festnetzteil der Ende-zu-Ende-Verbindung verwendet, während hingegen für die Funkübertragung ein daran angepaßtes *Last Hop Protocol* (LHP) genutzt wird. Das LHP ist einerseits für die Paketverlustcharakteristik des Wireless LAN optimiert, andererseits verfügt es über eine geringere Komplexität als das TCP, so daß das mobile Endgerät effektiv entlastet wird. Dabei ist das LHP so gehalten, daß die Ende-zu-Ende-Semantik der Anwendung erhalten bleibt.

Wie in Abb. 4.10 gezeigt, besteht die zweigliedrige Architektur aus dem *Local Socket Module* auf dem mobilen Terminal und dem *Export Socket Server* auf der Basisstation. Am entfernten Endgerät sind hingegen keine Änderungen erforderlich. Die Anwendungen des Terminals nutzen statt der regulären TCP-Aufrufe die äquivalenten Aufrufe des *Local Socket Module*, die dann von selbigem in das Exportprotokoll übersetzt werden. Der *Export Socket Server* leitet die Aufrufe dann an den eigentlichen TCP-Socket in der Basisstation weiter, der schließlich den Endpunkt für die TCP-Verbindung mit dem entfernten Endsystem darstellt. Obgleich nur für TCP umgesetzt, läßt sich das Prinzip der ReSoA auch auf andere Protokolle mit Socket-Schnittstellen wie beispielsweise UDP anwenden.

Die *ReSoA* ist, wie auch das vorherige Beispiel, ein Ansatz, bei dem verschiedene Anwendungsprotokolle entsprechend den Eigenschaften des Netzes unterschiedliche Implementierungen der entsprechenden Instanzen benötigen. Diese Dienste als programmierbare Dienste zu realisieren

bietet sich somit an, da sie von den Terminals selbst dann genutzt werden können, wenn sie nicht vorab installiert sind.

4.5 Konfigurieren und Steuern programmierbarer Dienste

Im einfachsten Fall besteht ein Netzdienst aus einem einzigen Teildienst, der einen – z. B. durch IP-Adressen von Sender und Empfänger – eindeutig gekennzeichneten Datenstrom bearbeitet. Dabei lassen sich zwei Funktionen trennen, nämlich einerseits die eigentliche Bearbeitung der Pakete sowie das *Steuern* und *Konfigurieren* dieser Bearbeitung andererseits. Die Konfiguration eines programmierbaren Netzdienstes ist (wie die Konfiguration anderer verteilter Systeme auch) eindeutig bezeichnet durch den ausgeführten Code, dessen jeweiligen Ausführungsort, der durch die IP-Adresse gegeben ist, sowie den entsprechenden Parametern.

4.5.1 Feature Interaction

Beeinflussen sich zwei (Teil-)Dienste in unerwünschter Art und Weise spricht man von *Feature* beziehungsweise *Service Interaction*, wobei die Unterscheidung hier nicht wesentlich ist. Allgemein bezeichnet man als *Feature Interaction* den Fall, daß das Verhalten des Gesamtsystems nicht den einzelnen Spezifikationen genügt. *Feature Interaction* kann auftreten, wenn die Beschreibung eines Systems durch seine Leistungsmerkmale unvollständig, uneindeutig oder fehlerhaft ist [Zav93, KSM00].

In einem programmierbaren System könnte beispielsweise folgende Situation auftreten: Ein von einem Administrator gestarteter Dienst zählt alle durchlaufenden Pakete, die bestimmten Kriterien genügen. Gleichzeitig ändert ein von einem Endgerät veranlaßter Dienst die Adresse der Datagramme, etwa um sie an den neuen Aufenthaltsort zuzustellen. Wird nicht genau festgelegt, welcher Dienst zuerst Zugriff auf die Pakete erhält, kann es zu unterschiedlichen Zählergebnissen kommen.

In der Telekommunikation gibt es nach [Fri96] drei Kategorien von Techniken um *Feature Interaction* zu begegnen:

Vermeiden *Feature Interaction* läßt sich vermeiden, indem bereits beim Entwurf eines Dienstes möglichen Konflikten begegnet wird, beispielsweise im Hinblick auf die Nutzung exklusiver Ressourcen. Im oben genannten Beispiel wäre der Zugriff auf bestimmte Pakete die zu koordinierende Ressource. Die Vermeidung würde dann mit sich bringen, daß keine zwei zur Verfügung gestellten Dienste sich überschneidende Paketselektionskriterien aufweisen dürfen. Diese sehr restriktive Strategie bringt es jedoch mit sich, daß jeder verfügbare Dienst bedenkenlos geladen werden kann, da *Feature Interaction* per se ausgeschlossen ist. Allerdings können dabei auch durchaus nützliche Anwendungen verhindert werden.

Entdecken Vor dem Starten eines Dienstes läßt sich *Feature Interaction* durch eine Analyse der formalen Dienstbeschreibungen hinsichtlich der auftretenden Zustandsübergänge entdecken. Dabei können Verklemmungen, Schleifen, nichtdeterministisches Verhalten, Zustandsübergänge zu abnormen Zuständen und Doppeldeutigkeiten erkannt werden [Fri96]. Da diese Analyse sehr aufwendig sein kann, sollte sie bereits vorab durchgeführt werden um bei Start des Dienstes zur Verfügung zu stehen.
Im oben genannten Beispiel würde entsprechend beim Laden geprüft, ob ein Dienst gleiche oder sich überlappende Paketselektionskriterien wie ein bereits aktiver Dienst anfordert. Ist dies der Fall so wird entweder der neue Dienst blockiert oder der bestehende Dienst zuvor terminiert. Diese Strategie ist weniger restriktiv als die der Vermeidung, da auf diese Weise potentiell interagierende Dienste verfügbar sind. Allerdings werden Dienste hierbei vorbeugend blockiert, auch dann wenn im Betrieb keine *Feature Interaction* aufgetreten wäre.

Auflösen Tritt *Feature Interaction* während der Laufzeit eines Dienstes zu Tage, so kann von Seiten der Dienststeuerung versucht werden, diese aufzulösen. Dazu kann sich die Dienststeuerung zunächst um eine Rekonfiguration bemühen. Für das eingangs genannte Beispiel würde dies bedeuten, daß die Dienste anhand einer zu ermittelnden Präzedenz die Datagramme erhalten würden. Erst wenn dies fehlschlüge, müßte eine Auswahl getroffen und einer der beiden Dienste blockiert werden.

Neben dem eigentlichen Dienstmodul muß daher in einem programmierbaren System immer auch ein Verzeichnis der Interaktionen mit anderen Dienstmodulen an zentraler Stelle verfügbar sein. Im folgenden Abschnitt wird abgeschätzt, wie groß die Zahl der zu berücksichtigenden Interdependenzen (wechselseitigen Abhängigkeiten) ist, um einen Anhaltspunkt für die Komplexität zu erhalten.

4.5.2 Abschätzung der Zahl der Interdependenzen bei Diensten

Setzt man voraus, daß Abhängigkeiten immer nur zwischen einzelnen Teildiensten bestehen, sind für n Teildienste insgesamt $\binom{n}{2} = \frac{n\cdot(n-1)}{2}$ Regeln erforderlich. Es können sich jedoch auch Gruppen von Teildiensten gegenseitig in einer Art und Weise beeinflussen, die nicht durch bilaterale Abhängigkeiten erfaßbar ist. Beispielsweise kann eine Funktionsgruppe bestehend aus zwei Elementen das Verhalten eines anderen Dienstes oder einer anderen Funktionsgruppe mit ebenfalls zwei Elementen beeinflussen. Prinzipiell ist jedes Paar von Diensten dahingehend zu untersuchen, ob es eine Funktionsgruppen bildet, so wie auch prinzipiell jedes Element von diesem beeinflußt werden könnte. Die Anzahl aller möglichen Interdependenzen zwischen Funktionsgruppen mit gleicher Elementzahl und solchen ungleicher Elementzahl beziehungsweise einzelnen Diensten läßt sich unter Berücksichtigung aller Permutationen wie folgt aufsummieren:

$$\underbrace{\binom{n}{2} + 3\cdot\binom{n}{4} + 10\cdot\binom{n}{6} + \cdots}_{\text{gleiche Elementezahl}} + \underbrace{3\cdot\binom{n}{3} + 4\cdot\binom{n}{4} + 10\cdot\binom{n}{5} + \cdots}_{\text{gemischte Elementezahl}} \tag{4.1}$$

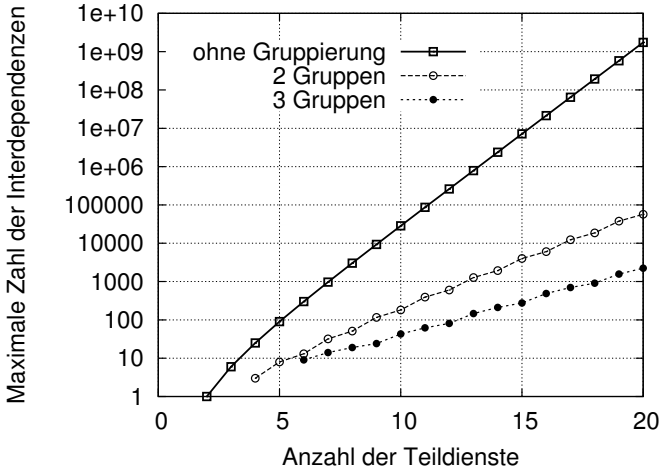


Abbildung 4.11: Maximale Anzahl möglicher Interdependenzen zwischen Teildiensten.

Daraus läßt sich folgender verallgemeinerter Ausdruck für die Anzahl N_{dep} der Interdependenzen zwischen n Diensten aufstellen:

$$N_{\mathrm{dep}}(n) = \sum_{k=1}^{\lfloor \frac{1}{2}n \rfloor} \frac{1}{2} \binom{2k}{k} \binom{n}{2k} + \sum_{k=2}^{n-1} \sum_{i=1}^{\min(k-1,n-k)} \binom{i+k}{k} \binom{n}{i+k} \tag{4.2}$$

Diese theoretische Zahl wird jedoch praktisch kaum erreicht, da es sehr unwahrscheinlich ist, daß alle Interdependenzen aller Ordnungen bei einem Netzdienst zugleich zutreffen. Dies senkt die Komplexität auf ein handhabbares Maß, jedoch sind bei der Spezifikation eines Netzdienstes grundsätzlich alle Möglichkeiten zu prüfen.

Ein strukturierter Ansatz, um die Komplexität dieses Vorgangs zu senken, ist die Bildung logischer Gruppen. Da $N_{\mathrm{dep}}(n)$ stark mit der Anzahl n der Elemente steigt, bewirkt eine Aufteilung in g Gruppen der Größe $G_1..G_g$ (wobei $\sum_{i=1}^{g} G_i = n$ ist) eine deutliche Reduktion:

$$N_{\mathrm{dep}}(n) \quad \ll \quad N_{\mathrm{dep}}(g) + \sum_{i=1}^{g} N_{\mathrm{dep}}(G_i) \tag{4.3}$$

Wie in Abb. 4.11 erkennbar ist, steigt die Zahl der möglichen Interdependenzen im einfachen Fall ohne Gruppierung stärker als exponentiell an. Für den Fall der Unterteilung in zwei – beziehungsweise drei – annähernd gleich große logische Gruppen ist die Anzahl der dann noch verbleibenden Interdependenzen ebenfalls in Abb. 4.11 eingezeichnet. Diese Reduktion der zu betrachtenden Interaktionen verkürzt zugleich den kritischen Pfad der Entscheidungsfindung und ermöglicht so eine schnellere Reaktion nach einem Handover des mobilen Teilnehmers. Die Funktionsgruppen können dabei sowohl anhand von topologischen (horizontalen) als auch funktionalen (nach ISO/OSI-Schicht, also vertikalen) Kriterien gebildet werden.

4.6 Flexible Signalisierung für programmierbare Dienste

Ein programmierbarer Netzdienst kann aus einer Vielzahl unterschiedlicher, nicht vorab bestimmbarer Teildienste zusammengesetzt sein. Daher ist es erforderlich, daß alle Dienstmodule eine einheitliche, aber möglichst universell verwendbare und erweiterbare Schnittstelle für die Dienstsignalisierung verwenden. Diese Schnittstelle muß dabei auf verschiedenen Plattformen verfügbar sein und auch auf weniger leistungsfähigen Rechnern mit verringertem Befehlsumfang lauffähig sein. Ferner soll die Kommunikation auch mit mobilen Teilnehmern und Diensten möglich sein. Die sogenannten Web Services stellen eine Lösung dar, die alle diese Voraussetzungen erfüllt.

4.6.1 Web Services

Die üblichste Definition für *Web Services* ist die des WWW-Konsortiums [HB04]:

> *Ein* Web Service *ist ein Softwaresystem, das die interoperable Interaktion zwischen unterschiedlichen verteilten Automaten über ein Netz ermöglicht. Seine Schnittstelle ist in einem maschinenlesbaren Format beschrieben (insbesondere WSDL). Andere Systeme interagieren mit dem Web Service entsprechend dieser Beschreibung über SOAP-Nachrichten, welche typischerweise XML-codiert mittels des HTTP und unter Verwendung weiterer Web-Standards transportiert werden.*

In [KL04] werden vier Eigenschaften identifiziert, durch die sich *Web Services* auszeichnen: *Lose Kopplung*, *Virtualisierung*, *einheitliche Konventionen* und (offene) *Standards*. Unter *loser Kopplung* versteht man, daß die Dienste voneinander unabhängig entwickelt werden können und ausschließlich durch die definierten Nachrichten gekoppelt sind. Die dynamische Bestimmung von Kommunikationspartnern durch eine Zwischeneinheit (*Middleware*) wird unter dem Begriff *Virtualisierung* zusammengefaßt und erlaubt eine weitergehende Abstrahierung der Dienste. Statt fest vordefinierter Protokollstrukturen gibt es einheitliche Konventionen die festlegen, wie passende Protokolle, Datenformate, etc. zu definieren sind.

4.6.2 Common Object Request Broker Architecture (CORBA)

Im Gegensatz zur losen Kopplung der *Web Services* steht die feste Kopplung bei Systemen wie beispielsweise CORBA [Obj04a] und *Remote Procedure Call* (RPC) [Sri95]. Wie in Abb. 4.12 gezeigt, werden bei CORBA alle Aufrufe durch den Object Request Broker gelenkt. Die kommunizierenden Instanzen müssen dabei nicht den Ort des Gegenübers kennen. Diese Virtualisierung wird dadurch erreicht, daß jeder Prozeß eine eindeutige Referenz besitzt, mittels derer er von jedem anderen Prozeß aufgerufen werden kann, unabhängig davon ob dieser auf dem gleichen oder einem anderen Rechner ausgeführt wird [Vin97]. Dabei besteht dennoch eine strikte Kopplung

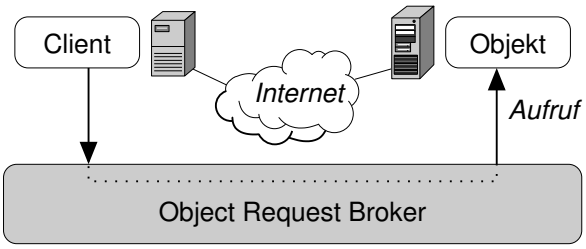


Abbildung 4.12: Virtualisierung der Objektzugriffe bei CORBA.

der Schnittstellen, die keinerlei Interpretationsspielraum läßt, wie er beispielsweise bei paralleler Verwendung unterschiedlicher Versionen hilfreich sein kann.

Verglichen zu anderen Systemen wie CORBA beruhen *Web Services* auf dem zustandslosen Austausch von *Simple Object Access Protocol* (SOAP)-Nachrichten [Mit03]. Durch die dabei verwandte XML-Codierung der Daten lassen sich besonders einfach zueinander kompatible Schnittstellen aufrechterhalten, da beim Parsen fehlende Parameter und unbekannte Felder entweder ignoriert, auf Standardwerte gesetzt oder mit weitergehenden Fehlerroutinen gezielt behandelt werden können.

Ferner ist CORBA nicht für mobile Umgebungen geeignet, insbesondere läßt sich keine für die Anwendung transparente Mobilitätsunterstützung realisieren. Mit Wireless CORBA [Obj04b] wurde zwar eine Möglichkeit spezifiziert, Mobilität als einen weiteren Freiheitsgrad neben der Virtualisierung einzuführen. Jedoch wird in [GKS02] konstatiert, daß einerseits noch keine Implementierung verfügbar sei und andererseits auch die minimale CORBA Spezifikation noch hohe Anforderungen an kleine mobile oder eingebettete Systeme stelle.

Im Umfeld mobiler programmierbarer Dienste überwiegen somit die Vorzüge von *Web Services*. Die zustandslose Kommunikation erlaubt besonders leichte Mobilitätsunterstützung. Andererseits ist die Schnittstellenspezifikation nicht starr, so daß auch in ihrer Spezifikation leicht voneinander abweichende Dienste miteinander kommunizieren können. Da die Schnittstellenspezifikation in maschinell auswertbarer Form vorliegt, können programmierbare Dienste diese bei Bedarf laden (beispielsweise von einem zentralen Server) und sind dann in der Lage ihre Signalisierungsfähigkeiten dynamisch zu erweitern.

4.7 Abstrahiertes Modell einer programmierbaren Plattform

Eine vielseitige programmierbare Plattform ist die Basis für die Realisierung transparenter Netzdienste mittels eines programmierbaren Netzes. Handelt es sich dabei um einen verteilten Dienst, so besteht dieser aus mehreren kommunizierenden Modulen. Daneben kann auch ein Dienst auf einem Knoten in mehrere Module aufgeteilt werden. Dies bedeutet zwar zunächst einen höheren Aufwand, erlaubt aber eine Wiederverwendung von Funktionen bei anderen Diensten und verbessert auch die Handhabbarkeit der Komplexität des Systems.

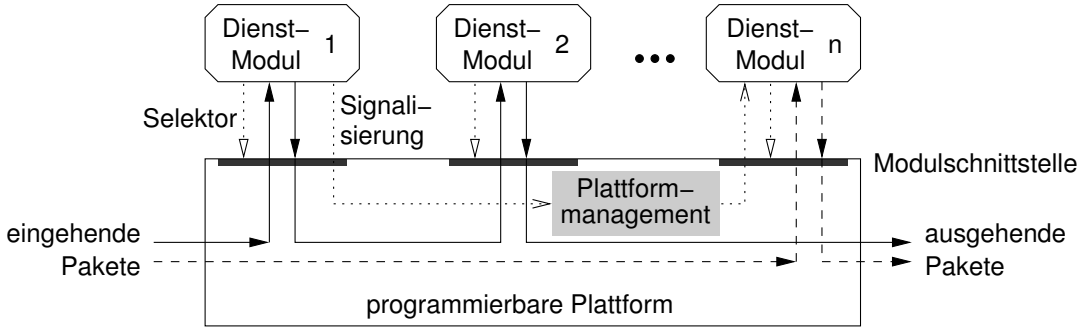


Abbildung 4.13: Vereinfachtes Modell einer modularen programmierbaren Plattform.

Die beiden wesentlichen Aufgaben der Plattform sind dabei die Möglichkeit des Zugriffs auf die durch den Knoten gerouteten *Datenpakete* einerseits und die *Kommunikation* mit anderen Dienstmodulen sowie weiteren Diensten andererseits. Für beides stellt die programmierbare Plattform entsprechende Dienstprimitive bereit. Im einfachsten Fall können Pakete, die bestimmte Kriterien erfüllen, dem Dienstmodul zur Bearbeitung übergeben und anschließend wieder von diesem übernommen werden. Hier beschränkt sich die Schnittstelle zwischen den Dienstmodulen und der programmierbaren Plattform auf die Übergabe der *Selektionskriterien* einerseits und der *Pakete* des Datenstromes andererseits, wie in Abb. 4.13 schematisch gezeigt. Die programmierbare Plattform muß dabei darauf achten, daß ein Netzdienst nicht unerlaubterweise auf Datenpakete anderer Nutzer zugreifen kann. Ferner sollte er nicht ohne Weiteres Datenströme selektieren können, die das Management der programmierbaren Plattform betreffen.

Darüber hinaus können bei mächtigeren Plattformen auch bestimmte weitergehende Bearbeitungsschritte von der programmierbaren Plattform übernommen und entsprechend den Anweisungen des Dienstmoduls ausgeführt werden. Es scheint dabei zunächst von untergeordneter Bedeutung, ob die Logik und Funktionalität der Paketbearbeitung in der *Middleware* oder der Anwendung selbst angesiedelt wird. Allerdings kann die programmierbare Plattform bestimmte Paketverarbeitungsschritte wiederum an das Betriebssystem übergeben, so daß diese direkt dort ausgeführt werden. Dies erhöht die Verarbeitungsgeschwindigkeit, da die Übergabe der Datenpakete über die *Middleware* an die Anwendung und wieder zurück entfällt.

Das Laden und Starten von Diensten im Allgemeinen und Dienstmodulen im Speziellen kann dabei auf verschiedene Arten und Weisen erfolgen. Prinzipiell ist immer ein Basisdienst mit einer entsprechenden, dem Endterminal vorab bekannten, Signalisierungsschnittstelle auf den programmierbaren Netzknoten erforderlich. Dieser reagiert auf die wie auch immer geartete Anforderung (siehe Abschnitt 4.3.4) für das Initiieren eines neuen Dienstes. Im Sinne der Definition in Abschnitt 4.3.2 handelt es sich dabei auch um einen Netzdienst, der selbstverständlich ebenfalls modular aufgebaut und dynamisch ladbar sein kann. Fester Bestandteil der programmierbaren Plattform ist hingegen eine Managementschnittstelle, die die Kontrolle aller Dienstmodule gestattet.

4.8 Zusammenfassung

In diesem Kapitel werden zuerst die Begriffe *Aktives Netz* und *Programmierbares Netz* in Anlehnung an die Literatur definiert. Anhand dreier repräsentativer Systeme werden dann die Aspekte der Codeverteilung und -ausführung einschließlich der sich daraus ergebenden Aspekte der Sicherheit dargelegt. Nach einer kurzen Erläuterung einiger Architekturen für aktive und programmierbare Systeme wird auf die mittels derartiger Systeme erbrachten Dienste eingegangen.

Aufbauend auf dem Konzept des *Netzdienstes* werden Aspekte der Leistungsmerkmale, Initiierung von Diensten und Service Discovery in Aktiven und Programmierbaren Netzen diskutiert. Verdeutlicht werden diese Aspekte bei der folgenden Darstellung einiger aktiver beziehungsweise programmierbarer Dienste. Bedingt durch die höhere Dynamik der Dienste stellen sich Fragen der *Feature Interaction* in Aktiven und Programmierbaren Netzen in stärkerem Maße als in herkömmlichen Netzen. Die sich ergebenden Problemstellungen und Lösungsansätze zur Konfiguration und Steuerung werden anhand einer Komplexitätsabschätzung erläutert. Nach einer Würdigung der beiden wichtigsten strukturierten Ansätze zur Signalisierung im Kontext der Programmierbarkeit wird abschließend ein idealisiertes Modell einer programmierbaren Plattform vorgestellt, welches als Basis für die im nächsten Kapitel entworfene Architektur für mobile programmierbaren Netzdienste dienen soll.

Kapitel 5

Flexible Architektur für mobile programmierbare Dienste

Dienste in einem mobilen Umfeld zu erbringen birgt einige besondere Herausforderungen. Führt der Teilnehmer einen nicht antizipierten Handover durch, bricht die Datenanbindung plötzlich ab. Nach einer bestimmten Latenzzeit, während der keinerlei Verbindung besteht, verbindet sich der Teilnehmer mit einem im Allgemeinen nicht eindeutig vorherbestimmbaren Netzzugangspunkt. Er erwartet aber, daß alle von ihm zuvor genutzten Dienste mitsamt den individuellen Einstellungen sofort wieder verfügbar sind.

Im folgenden wird zunächst die Mobilität von Diensten untersucht und darauf aufbauend eine *Architektur* zur transparenten Unterstützung mobiler programmierbarer Dienste entworfen. Dabei wird angenommen, daß programmierbare Plattformen in den Zugangsdomänen vorhanden sind, nicht jedoch im Bereich der Transport- beziehungsweise Kernnetze. Die hier vorgestellte Architektur ermöglicht es, den Dienstkontext an den neuen Aufenthaltsort zu transferieren oder die Anwendungsdaten temporäre dorthin weiterzuleiten. Schließlich wird ein heuristischer *Algorithmus* zur Auswahl der jeweils effizienteren Variante entworfen und mit dem theoretischen Optimum verglichen.

5.1 Dienste und Mobilität

In einem einfachen Szenario bewegt sich ein mobiler Teilnehmer zwischen den Versorgungsgebieten mehrerer *Wireless LAN*-Basisstationen. Die Mobilität des Teilnehmers wird durch eine *Mobile IP*-Infrastruktur unterstützt. Zusätzlich sind auf den Foreign Agents auch programmierbare Plattformen vorhanden, die das Ausführen programmierbarer Dienste ermöglichen. Dort wird für den Teilnehmer ein solcher programmierbarer Dienst ausgeführt, wie beispielsweise ein Protokollheader-Komprimierer für ein bestimmtes von ihm genutztes Protokoll. Dieser Dienst wird auf die Anforderung des Teilnehmers hin geladen – beispielsweise aus einem zentralen Repositorium – und gemäß seinen Anforderungen konfiguriert.

Das bereits in Abschnitt 2.2 vorgestellte Mobile IP wird für den Teilnehmer nach einem *Handover* lediglich die Zustellung der Datagramme von und zu seinem neuen Aufenthaltsort sicherstellen. Dies geschieht für die Anwendungen – abgesehen von der kurzen Unterbrechung der Datenübertragung – vollkommen transparent. Die Anwendungsprogramme des Teilnehmers können anschließend mit ihrer Kommunikation fortfahren, ohne ihre Transportverbindungen erneut aufbauen zu müssen.

Für *programmierbare Dienste* bietet Mobile IP jedoch keine Unterstützung, so daß diese nach dem Handover des Terminals am vorherigen Netzzugangspunkt verbleiben – und dort weiterhin Ressourcen belegen –, am neuen Ort jedoch fehlen. Die entsprechende Anwendung auf dem Terminal wird früher oder später erkennen, daß der Dienst nicht mehr verfügbar ist. Das Anwendungsprogramm wird daraufhin den zuvor genutzten Dienst am neuen Ort neu instantiieren. Die verwaiste Dienstinstanz am alten Netzzugangspunkt wird hingegen erst nach einer Zeitüberschreitung beendet.

5.1.1 Kontexttransfer

Um eine raschere Wiederaufnahme eines Dienstes nach einem Handover zu ermöglichen, wurde in [KP01b] der *Kontexttransfer* vorgeschlagen. Dabei wird der Kontext von Diensten vom alten zum aktuellen Netzzugangspunkt übertragen. Der Begriff *Kontext* wird dabei in [Kem02] folgendermaßen definiert:

> *Diejenige Information über den augenblicklichen Zustand eines Dienstes die erforderlich ist, um den Dienst in einem neuen Subnetz wiederherzustellen ohne dazu den gesamten Protokollaustausch mit dem mobilen Terminal nochmals durchzuführen.*

Der *Kontexttransfer* findet dabei stets zwischen zwei gleichartigen Dienstinstanzen statt, setzt also das Vorhandensein einer solchen am neuen Netzzugang voraus. Der Kontext läßt sich unterteilen in einen *statischen* und einen *dynamischen* Anteil. Der statische Kontext umfaßt die Parameter des zu bearbeitenden Datenstroms, die dem Dienst vorgegebenen Parameter und statische Zustandsvariablen der Paketverarbeitung.

Bei der *Robust Header Compression* (RoHC) [BBD+01] ist beispielsweise der Datenstrom durch IP-Adressen und Portnummern gekennzeichnet, das Komprimierungsschema ist ein vorzugebender Parameter. Dies sind die weitgehend *statischen* Zustandsparameter, die im sogenannten *First Order Context* zusammengefaßt werden. Diese Werte ändern sich nur selten, beispielsweise wenn der Benutzer die Konfiguration des Dienstes modifiziert. Der *dynamische* Anteil besteht im Gegensatz dazu aus Zustandsvariablen, die sich mit der Bearbeitung des Datenstromes ändern. Diese sind im sogenannten *Second Order Context* zusammengefaßt. Im Falle der Protokollheader-Komprimierung wären dies beispielsweise die aktuellen Werte der komprimierten Protokollheaderfelder. Der Zustand eines Dienstes wird nach einem Handover des Terminals mittels eines Kontexttransfers an eine Instanz des gleiche Dienstes am neuen Netzzugangspunkt übertragen, wie in Abb. 5.1 schematisch gezeigt.

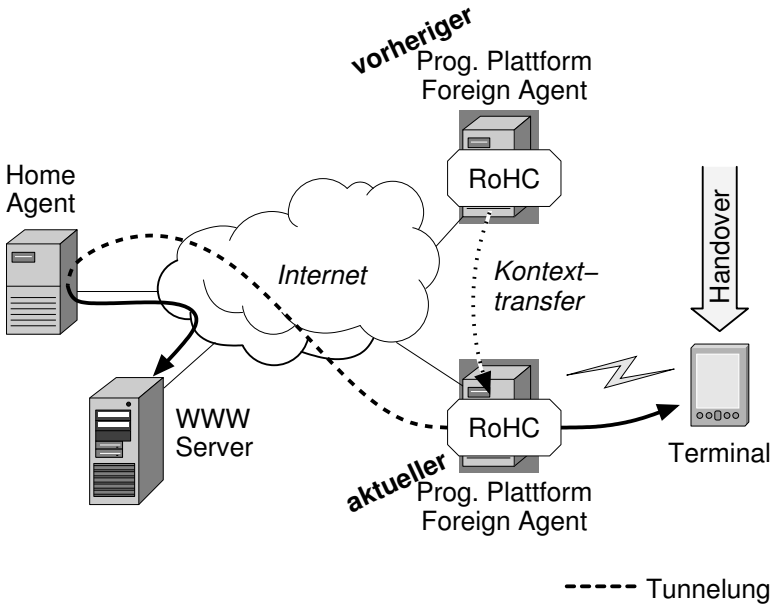


Abbildung 5.1: Kontexttransfer für die RoHC nach einem Handover.

Indirekte Transportansätze wie beispielsweise *I-TCP* [BB97] und *M-TCP* [BS97] trennen die Transportverbindungen des Terminals an einer Mobilitätsunterstützungsentität auf. Auf diese Weise werden Sendewiederholungen verlorener Datagramme sowie die Flußkontrolle separat auf den drahtlosen und den fixen Teil der Übertragung angewandt. Im Falle eines Handover werden die Zustände der Mobilitätsunterstützungsentität mittels eines Kontexttransfers auf die entsprechende Einheit im neuen Zugangsnetz übertragen, so daß die Verbindung dort weitergeführt werden kann. Wurden der entfernten Gegenstelle Segmente bestätigt, bevor deren Empfang vom mobilen Teilnehmer quittiert wurde, so muß dabei der vollständige Pufferinhalt mitgeführt werden. Andernfalls kann dieser im Vertrauen auf die Sendewiederholung des TCP verworfen werden.

Für den Kontexttransfer gelten dabei von der jeweiligen Anwendung abhängige strikte Zeitbeschränkungen. Im Falle der Protokollheader-Komprimierung verlieren die Zustände von Komprimierer und Dekomprimierer mit jedem Datagramm, welches nicht von beiden Entitäten bearbeitet wird, ein Stück ihrer Synchronisation. Dauert also der Kontexttransfer zu lange, ist der dynamische Kontext bereits veraltet und somit wertlos, da sich Komprimierer und Dekomprimierer ohnehin neu synchronisieren müssen. Lediglich den quasi-statischen *First Order Context* zu übertragen hätte in diesem Falle folglich den gleichen Effekt.

Mit dem *Context Transfer Protocol* (CXTP) wurde von der IETF-Arbeitsgruppe *Seamoby* (Seamless Mobility) in [LNPK05] ein Protokoll für den Kontexttransfer standardisiert. Dieses ist darauf ausgelegt, daß der Kontexttransfer vom mobilen Teilnehmer stets selbst initiiert wird. In Abb. 5.2 ist ein Nachrichtenflußdiagramm für einen Kontexttransfer nach einem *break-before-make* Handover gezeigt.

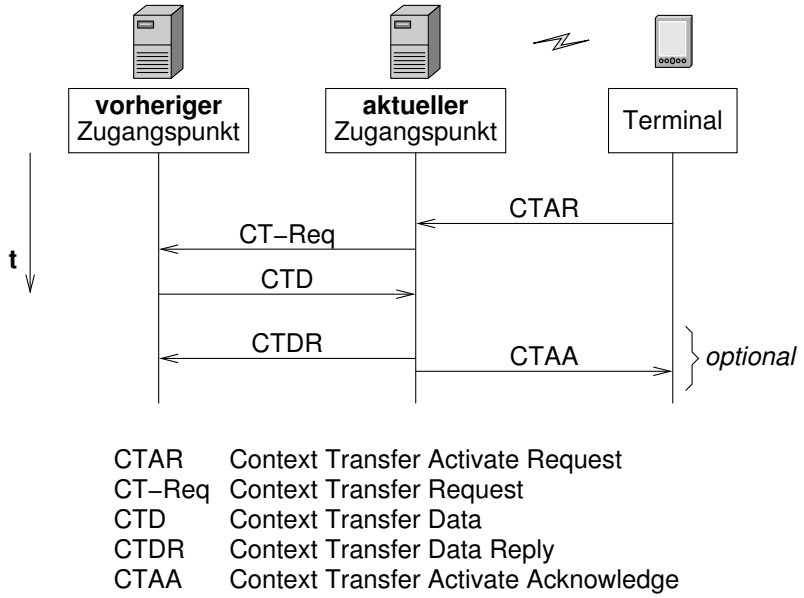


Abbildung 5.2: Nachrichtenflußdiagramm des CXTP nach einem Handover.

Charakterisierend für das CXTP ist, daß die Steuerung vollständig vom Terminal ausgeht. Es werden somit nur diejenigen Dienstkontexte transferiert, die vom Terminal explizit angefordert wurden. Folglich sind Kontexttransfere für transparente Dienste ausgeschlossen, die nicht vom Terminal selbst, sondern beispielsweise von einem Administrator initiiert werden.

5.1.2 Kollokation von programmierbarer Plattform und Foreign Agent

Führt ein Dienst eine zustandsbehaftete Bearbeitung der Pakete eines Datenstromes aus, so ist es unerläßlich, daß auch tatsächlich *alle* Pakete eines Datenstromes die betreffende programmierbare Plattform passieren. Dies ist dann nicht sichergestellt, wenn es mehrere alternative Pfade gibt, von denen mindestens einer die programmierbare Plattform mit dem Dienst umgeht. Verfügt also beispielsweise ein Netz über mehrere Gateways, so ist es nicht ohne weiteres möglich, einen zustandsbehafteten Dienst dort zu platzieren.

Wie in Abb. 5.3 dargestellt, ist der *Foreign Agent* der Dreh- und Angelpunkt aller Datenströme eines mobilen Teilnehmers, solange dieser in einem IP-Netz verweilt. Siedelt man den Dienst wie in [**TB02**] vorgeschlagen dort an, so können alle Datenpakete des mobilen Teilnehmers bearbeitet werden. Selbst wenn es mehrere Foreign Agents in einem Netz gibt, wird nicht zu einem anderen Foreign Agent gewechselt, solange dieser funktionsfähig ist.

Das Abkoppeln des Dienstes von den Basisstationen bringt mehrere Vorzüge mit sich. Einerseits wechselt ein Terminal häufiger seine Basisstation als das IP-Netz, so daß der Dienst in geringerem Maße von der Mobilität des Teilnehmers betroffen ist. Andererseits bleibt das System offen für zukünftige – unter Umständen proprietäre – Verbesserungen auf der Sicherungsschicht, wie beispielsweise durch das *Inter Access Point Protocol* (IAPP) [IEE03]. Nicht zuletzt sind Wireless

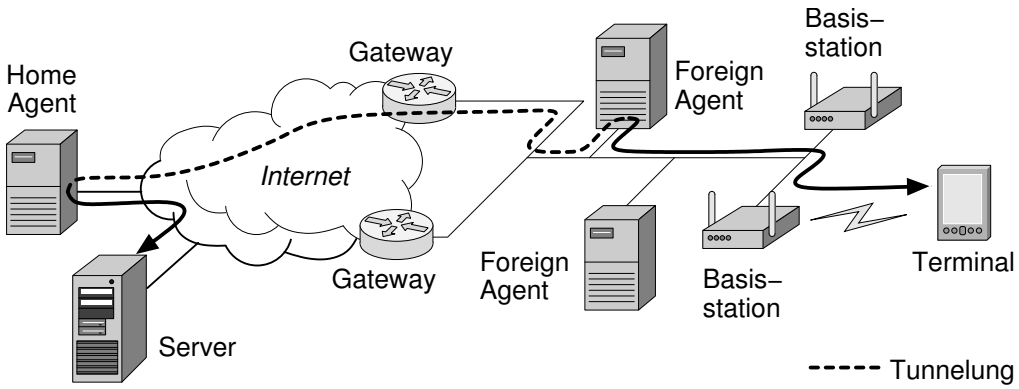


Abbildung 5.3: Datenpfad im Falle eines Netzes mit jeweils zwei Basisstationen, Foreign Agents und Gateways.

LAN-Basisstationen inzwischen zu kostengünstigen hochintegrierten Geräten geworden, die in großen Stückzahlen gefertigt werden und sich daher nur schwerlich mit besonderen oder aufwendigen Zusatzfunktionen ausstatten lassen.

5.1.3 Abgrenzung von Terminal- und Dienstmobilität

Das Ziel der hier vorgeschlagenen Architektur ist die transparente Mobilität von Terminals und Diensten. Dabei umfaßt die *Terminalmobilität* die Funktionalität von Mobile IP, also die transparente Mobilitätsunterstützung auf IP-Ebene. Allerdings ist die Architektur dahingehend universell, daß statt Mobile IP auch ein anderer Mechanismus eingesetzt werden kann.

Unter *Dienstmobilität* ist zunächst allgemein zu verstehen, daß ein netzseitiger Dienst bewegt wird. Dienst- und Terminalmobilität stehen orthogonal zueinander, da ein Dienst auch unabhängig von der Mobilität des Terminals verlagert werden kann. Dabei wird jedoch im Sinne der Definition in Abschnitt 4.3 angenommen, daß ein Dienst (genauer: eine Dienstinstanz) stets nur von einem einzigen Terminal genutzt wird[1].

Das Bindeglied zwischen einem Dienst und dem ihn nutzenden Terminal ist ein datenstromspezifisches Routing. Mittels dieses höherschichtigen Routings läßt sich dafür Sorge tragen, daß der richtige Datenstrom vom richtigen Terminal zum richtigen Dienst – und umgekehrt – gelenkt wird.

5.1.4 Anforderungen an eine Architektur für Dienstmobilität

Aus den in den dargelegten Randbedingungen lassen sich mehrere Anforderungen an eine Architektur zur Unterstützung der Mobilität von transparenten programmierbaren Diensten ableiten:

[1] Verschiedene Terminals können ungeachtet dessen jeweils eigene Instanzen des gleichen Dienstes nutzen.

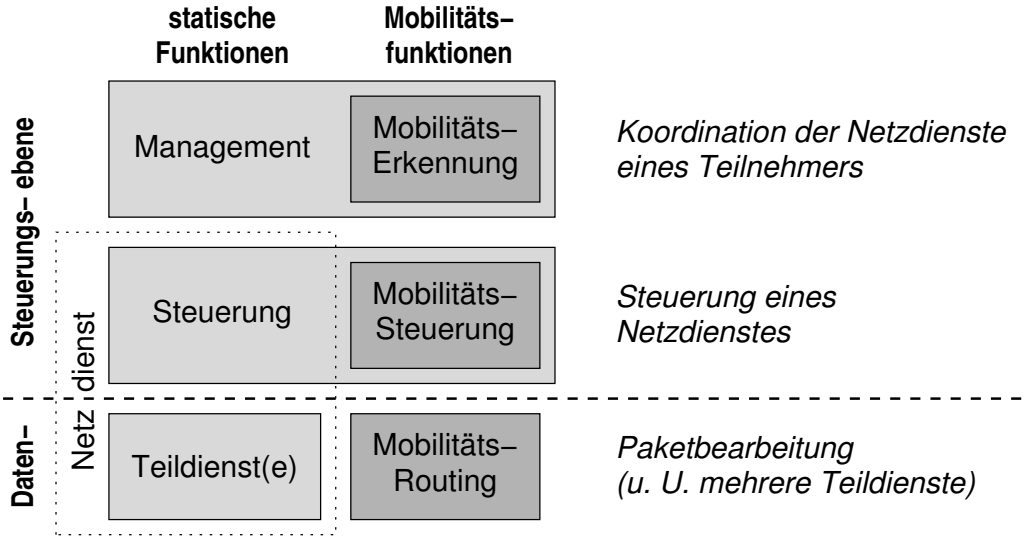


Abbildung 5.4: Hierarchie der Elemente der modularisierten Architektur.

Geschwindigkeit Die Dienste müssen nach einem Handover rasch wieder zur Verfügung stehen. Wie oben exemplarisch für die Headerkompression ausgeführt, nimmt der Performanzvorteil des Diensttransfers gegenüber einer Neueinrichtung mit fortschreitender Zeit ab.

Kompatibilität Die Architektur muß mit den Entitäten der Mobilitätsunterstützung kompatibel sein. Beispielsweise sollte bei Mobile IP die programmierbare Plattform auf dem Foreign Agent platziert sein, um alle Datagramme der Teilnehmer bearbeiten zu können.

Erweiterung des Kontexttransfers Beim bisherigen Modell ist ein Kontexttransfer zwischen *gleichartigen*, bereits vorhandenen Diensten vorgesehen. Außerdem ist er vom Teilnehmer selbst zu veranlassen. Zur Erbringung transparenter programmierbarer Dienste für mobile Teilnehmer ist daher eine Erweiterung zum Diensttransfer nötig.

Unbewegliche Dienste Um universell einsetzbar zu sein, sollte die Architektur auch die Bereitstellung unbeweglicher oder nur mit übermäßigem Aufwand bewegbarer Dienste unterstützen, beispielsweise durch Datentunnelung.

5.2 Modularisierte Mobilitätsunterstützung für Netzdienste

Um eine möglichst große Flexibilität hinsichtlich der Erweiterbarkeit beziehungsweise Austauschbarkeit von Funktionen zu erhalten, wird die in Abb. 5.4 dargestellte *dreischichtige* hierarchische Architektur vorgeschlagen. Wie bereits in Abschnitt 4.3.2 dargelegt, besteht ein Netzdienst aus einem oder mehreren Teildiensten und einer zugehörigen Steuerungseinheit. Für mobile Terminals sind einige weitere Elemente erforderlich, die im folgenden vorgestellt werden. Die Elemente und ihre jeweilige Aufgabe sind in Tab. 5.1 nochmals kurz zusammengefaßt.

Tabelle 5.1: Aufgaben der einzelnen Elemente der Architektur.

Element	Aufgabe
Management	Laden von Diensten sowie Sammeln und Verteilen von Signalisierungsnachrichten von / an die Dienste.
Mobilitätserkennung	Erkennen und Signalisieren des neuen Netzzugangspunktes eines Teilnehmers.
(Mobilitäts-)Steuerung	Bereitstellen eines Dienstes am neuen Ort, z. B. mittels eines Kontexttransfers.
Mobilitätsrouting	Hilfsdienst zum Anpassen des Routings der Datenpakete zum mobilen Teilnehmer mittels Datentunnelung.

Die *Managementeinheit*, die auf jedem programmierbaren Knoten vorhanden ist, sorgt einerseits für das Laden und Starten von Diensten (mitsamt zugehöriger Steuerungseinheit) indem sie die Anfragen des Teilnehmers auswertet. Außerdem leitet sie Signalisierungsnachrichten zwischen den Dienststeuerungen der verschiedenen Netzdienste eines Teilnehmers weiter.

Die *Mobilitätserkennung* ist Teil des Managements und muß vorab in allen potentiellen Zugangsnetzen vorhanden sein, um den mobilen Teilnehmer dort detektieren zu können. Kommt Mobile IP zum Einsatz, so ist dafür keine zusätzliche Signalisierung erforderlich, vielmehr kann die Mobilitätserkennung die entsprechenden Informationen aus den Mobile IP-Signalisierungsnachrichten entnehmen und an die Steuerung der Dienste des Teilnehmers melden.

Die *Dienststeuerung* ist bei aus mehreren Komponenten zusammengesetzten Netzdiensten erforderlich. Da es jedem Teildienst jederzeit möglich sein sollte, mit der Dienststeuerung zu kommunizieren, sollte diese dauerhaft an einem festen Ort angesiedelt sein. Andererseits ist es aufgrund der Mobilität des Teilnehmers erforderlich, daß die Steuerung am jeweils aktuellen Aufenthaltsort des Teilnehmers sein sollte, um auf einen Handover schnell lokal reagieren zu können. In diesem Falle müßte jedoch jeder Teildienst von jedem Umzug der Steuerung benachrichtigt werden, was zusätzlichen Signalisierungsaufwand erzeugen würde. Diese beiden scheinbar entgegengesetzten Anforderungen lassen sich vereinen wenn man in Betracht zieht, daß es sich bei der Steuerung der Teildienste und der Reaktion auf Handover um weitgehend zueinander orthogonale Vorgänge handelt. Daher wird die Dienststeuerung *zentral* angeordnet und *delegiert* die mobilitätsrelevanten Funktionen an die vor Ort (*dezentral*) agierende Mobilitätssteuerung.

Wird ein Netzzugangspunktwechsel erkannt, transferiert die *Mobilitätssteuerung* die Dienste des mobilen Teilnehmers an den neuen Aufenthaltsort und/oder paßt das *Mobilitätsrouting* an. Das Mobilitätsrouting deckt die Funktionen oberhalb der durch Mobile IP bereitgestellten Basisfunktionalität ab, also die Weiterleitung von Datagrammen anhand von Kriterien der Transport- bis Anwendungsschicht. Um Dienste auf eine andere programmierbare Plattform zu transferieren bedarf es jedoch noch zusätzlicher Funktionalitäten, die über den Kontexttransfer hinausgehen.

Tabelle 5.2: Vergleich von Kontexttransfer und Dienstmobilität.

	Kontexttransfer	**Dienstmobilität**
Steuerung	Im Endgerät	Im Netz
Signalisierung des Terminals	Für jeden Dienst separat	Für alle Dienste zusammen
Transparente Dienste	–	✓

5.2.1 Erweiterung des Kontexttransfers zum Diensttransfer

Der Kontexttransfer (wie er von der IETF standardisiert wird) setzt voraus, daß am neuen Netzzugangspunkt eine gleichartige Instanz des Dienstes bereits vorhanden ist. Dies ist in einem programmierbaren Umfeld mit einer Vielzahl möglicher Dienste nicht notwendigerweise der Fall. Ferner sieht der Vorschlag der IETF vor, daß der Kontexttransfer vom *Terminal* initiiert wird. Dies impliziert aber, daß das Terminal vollständige Kenntnis über alle tatsächlich genutzten Dienste hat. Dies trifft beispielsweise bei *transparenten* Netzdiensten nicht notwendigerweise zu. Schlußendlich muß das Terminal die CTAR-Nachricht richtig adressieren können, was in allgemeinen Topologien nicht ohne Weiteres möglich ist.

Für den Bereich der mobilen programmierbaren Netzdienste wird daher der Begriff des *Diensttransfers* als eine relativ zum Kontexttransfer weiterreichende Methode der Mobilitätsunterstützung eingeführt. Ein vollständiger Diensttransfer besteht aus vier Arbeitsschritten: *Instantiieren* des Dienstes am neuen Ort, *Transfer* des statischen Kontextes, gegebenenfalls *Synchronisieren* des dynamischen Kontextes und falls erforderlich noch dem *Anpassen* der Konfiguration des Dienstes.

Wesentliche Eigenschaften des Diensttransfers sind die Unabhängigkeit von vorhandenen Instanzen, die Möglichkeit diesen ohne aktives Zutun des Terminals durchzuführen (z. B. für transparente Dienste) und die weitgehende Unabhängigkeit von den topologischen Gegebenheiten des Netzes. In Tab. 5.2 sind die Unterschiede von Kontexttransfer und Dienstmobilität nochmals zusammengefaßt.

Abhängig davon in welchem Zeitrahmen der Diensttransfer erfolgt – ob er dem Terminal nachfolgt, es begleitet, oder ihm vorauseilt – lassen sich folgende Fälle unterscheiden:

Reaktiver Transfer nach einem *break-before-make* Handover: Der Dienst wird am neuen Ort geladen (falls erforderlich), dann gestartet und dabei mit den transferierten statischen und eventuell auch dynamischen Kontextdaten initialisiert. Neben der durch den Handover bedingten Unterbrechung kommt hier noch die Zeit hinzu, die der Dienst zu seiner Rekonfiguration (beziehungsweise Resynchronisierung) benötigt.

Präemptiver Transfer als Maßnahme für einen bevorstehenden *make-before-break* Handover: Der Dienst wird am neuen Ort geladen und mit dem *statischen* Kontext initialisiert. Erfolgt schließlich der Handover, wird der Diensttransfer durch den *dynamischen* Kontext vervollständigt. Allerdings kann auch hier kein unterbrechungsfreier Diensttransfer garantiert werden, da der Dienst zumindest für die Zeit des Transfers des dynamischen Kontextes angehalten werden muß.

Administrativer Transfer an einen günstigeren Ort: Der Dienst wird am neuen Ort geladen und mit dem statischen Kontext initialisiert. Da diese Verlagerung im Gegensatz zum Handoverfall nicht zeitkritisch ist, kann die Umschaltung zwischen den Diensten erst dann erfolgen, nachdem der dynamische Kontext vollständig synchronisiert ist.

In Abb. 5.5 ist in einem Nachrichtenflußdiagramm der Aufbau und erstmalige Transfer eines Dienstes gezeigt. Der Übersichtlichkeit halber wurden die Bestätigungsnachrichten zur Entdeckung von Übertragungsfehlern sowie die Nachrichten zum Zwecke der Authentifizierung und Autorisierung der verschiedenen Instanzen untereinander weggelassen.

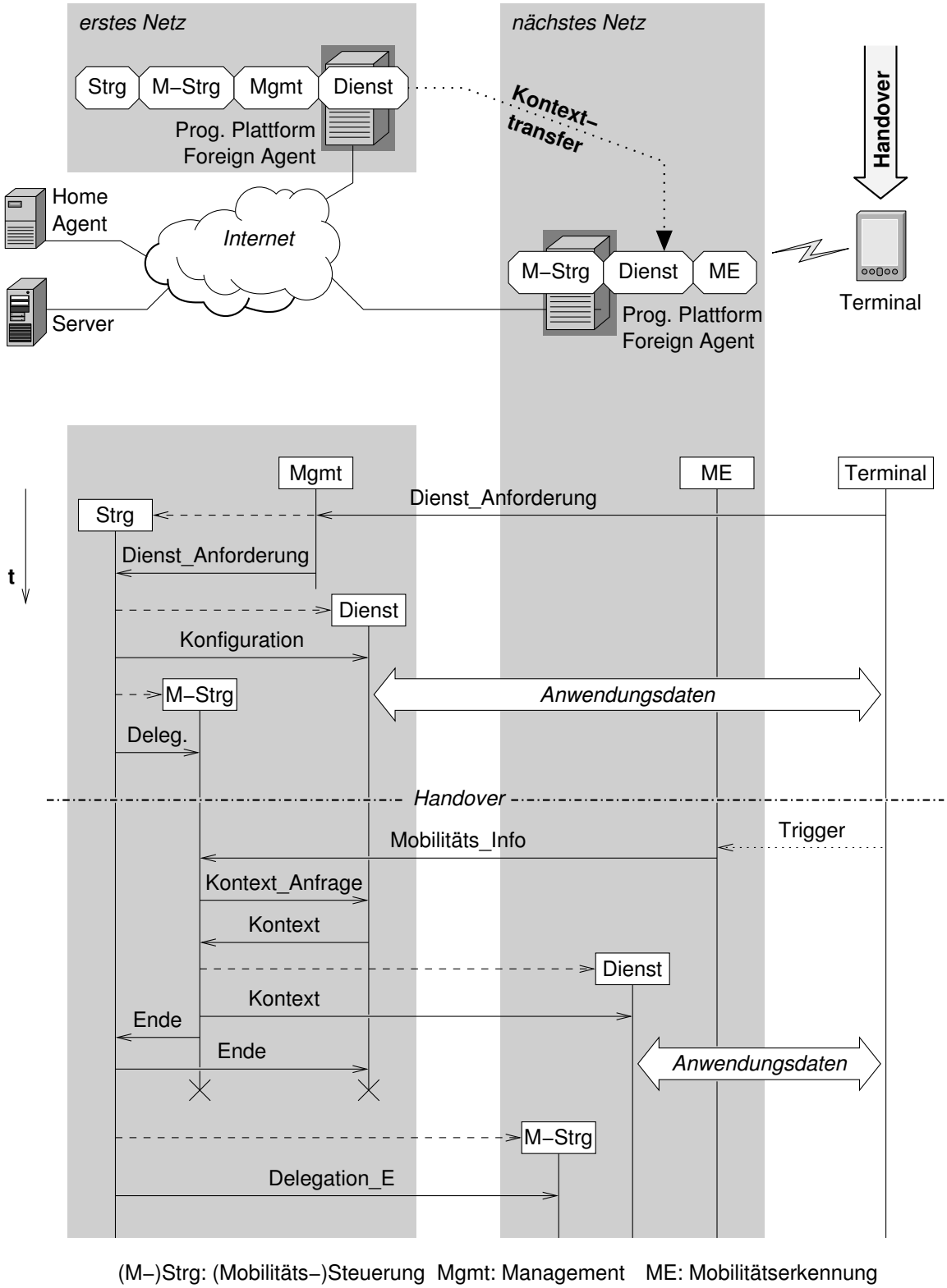
5.2.2 Bereitstellung immobiler Dienste durch Datentunnelung

Bei einem Diensttransfer wurde immer vorausgesetzt, daß sich der Dienst auch wirklich bewegen läßt. Es gibt jedoch auch Dienste, die sich nicht verlagern lassen oder bei denen ein Bewegen keinerlei Vorteil bringen würde. Die Gründe dafür können verschiedenster Natur sein:

- Der Kontext ist zu groß und ändert sich gleichzeitig auch rasch, wie dies beispielsweise bei der Transcodierung von Multimediadatenströmen der Fall ist
- Der Dienst läßt sich auf Grund vorhandener (z. B. Hardware-)Einschränkungen nicht am neuen Ort ausführen
- Es handelt sich um einen herkömmlichen statischen Dienst, der keinen Kontexttransfer unterstützt
- Für die beim Dienst eintreffenden Datenströme ist keine Mobilitätsunterstützung vorhanden, so daß die Datenströme weiterhin über den bisherigen Ort des Dienstes geleitet werden müssten

Will man einen solchen Dienst im Falle eines Handover nicht abbrechen lassen, kann alternativ auch das Mobilitätsrouting so angepaßt werden, daß der betreffende Datenstrom weiterhin vom Terminal an den Ort des Dienstes – und umgekehrt – geroutet wird, wie in Abb. 5.6 beispielhaft skizziert. Daneben kann es sich aber auch bei kurzlebigen Diensten als günstiger erweisen, Datenströme für die – kurze – Restdauer umzuleiten.

Die Datentunnelung ist jedoch nicht völlig unproblematisch. Zunächst ist es offensichtlich, daß die Tunnelung eine erhöhte Netzlast mit sich bringt, da die Datenströme einen längeren Weg



(M–)Strg: (Mobilitäts–)Steuerung Mgmt: Management ME: Mobilitätserkennung

Abbildung 5.5: Vereinfachtes Nachrichtenflußdiagramm eines Handover mit Diensttransfer.

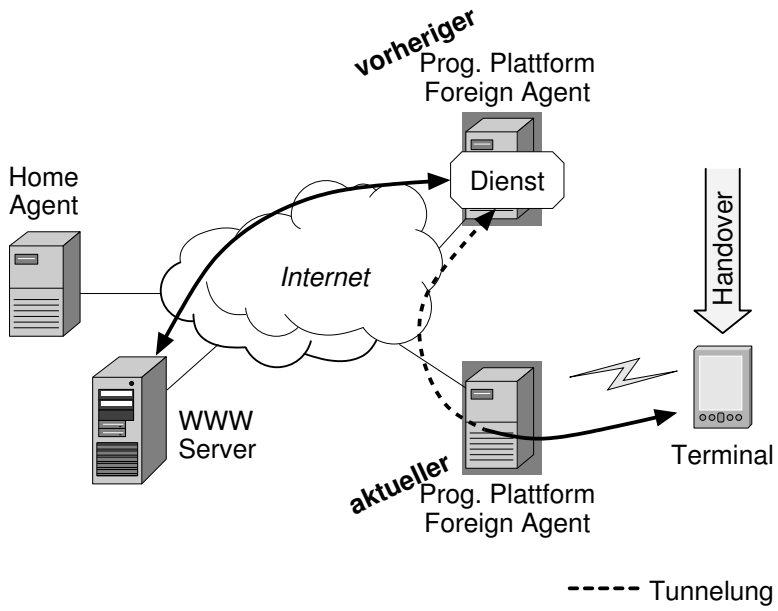


Abbildung 5.6: Bereitstellung des Dienstes nach einem Handover durch Datentunnelung.

in den Zugangsnetzen zurücklegen. Dies ist dann erheblich, wenn die Übertragungskapazität der Zugangsnetze in der Größenordnung der Summe der Datenraten der daran angeschlossenen Basisstationen liegt, so daß im Hochlastbetrieb die zusätzlich erzeugte Last eine Überlastung hervorruft.

Ferner müssen ausnahmslos alle Pakete der betreffenden Verbindung vom aktuellen Aufenthaltsort des Terminals zum vorherigen Netzzugangspunkt zurückgeroutet werden. Dies ist besonders kritisch im Falle eines transparenten Proxy (z. B. für HTTP), einer Adressübersetzung (NAPT) oder eines ähnlichen Dienstes. Erreicht auch nur ein IP-Paket den Server, ohne zuvor den Dienst zu passieren, wird ein TCP-Fehler ausgelöst, was einen Abbruch der Transportverbindung zur Folge hat. Konkret bedeutet das, daß Pakete eines mobilen Teilnehmers nach einem Handover erst weitergeleitet werden dürfen nachdem feststeht, daß dieser Fall des transparenten Proxy nicht vorliegt.

In Abb. 5.7 ist die Signalisierung zwischen den beteiligten Instanzen für den Fall des Dienst-transfers mittels *Datentunnelung* dargestellt. Auch hier wird die mit der Mobilität verbundene Funktionalität von der Dienststeuerung (**Strg**) an die Mobilitätssteuerung (**M-Strg**) delegiert. Nach einem Handover, der von der im neuen Netz vorhandenen Mobilitätserkennung (**ME**) gemeldet wird (die Erkennung könnte beispielsweise durch einen Trigger von Seiten des mobilen Terminals erfolgen), instantiiert die Mobilitätssteuerung im alten wie im neuen Netz eine Instanz des Mobilitätsroutings (**MR**) und konfiguriert diese entsprechend. Schlußendlich wird im neuen Netz eine neue Instanz der Mobilitätssteuerung instantiiert, an die dann die Dienststeuerung wieder die mit der Mobilität verbundenen Aufgaben delegiert.

Wie bereits in Abb. 5.5 wurden zum Zwecke einer klareren Darstellung die Bestätigungsnachrichten zur Entdeckung von Übertragungsfehlern sowie die Nachrichten zum Zwecke der Au-

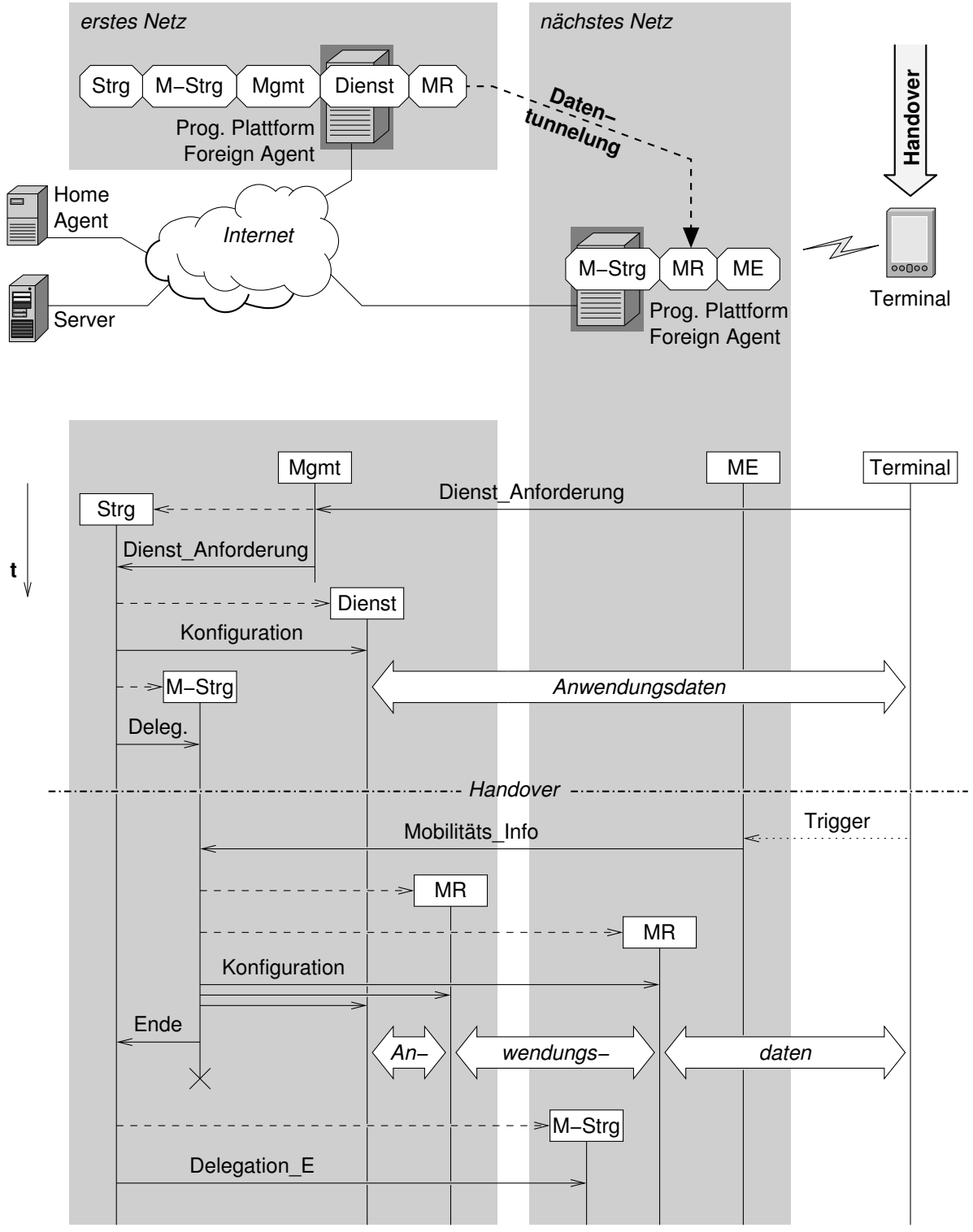


Abbildung 5.7: Vereinfachtes Nachrichtenflußdiagramm eines Handover mit Datentunnelung.

thentifizierung und Autorisierung der verschiedenen Instanzen untereinander weggelassen. Im Falle eines aus mehreren Teildiensten bestehenden Netzdienstes wäre in der Graphik jeweils *Dienst* durch *Teildienst* zu ersetzen. Die Steuerung würde dann die einzelnen Nachrichten an jeden dieser Teildienste senden. Analog würde das Management auch die Dienststeuerungen mehrerer Netzdienste verwalten, wenn der Teilnehmer gleichzeitig mehrere Netzdienste nutzte.

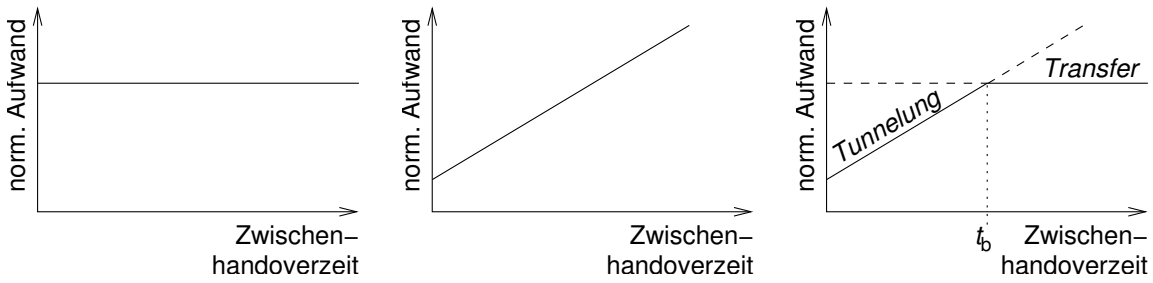
5.3 Diensttransfer oder Datentunnelung?

Wie oben bereits dargelegt, läßt sich die Dienstmobilität, also die Bereitstellung des Dienstes nach einem Handover am neuen Netzzugangspunkt des Teilnehmers, sowohl mit Hilfe eines Diensttransfers als auch einer Datentunnelung durchführen. In Abb. 5.8 werden diese beiden Methoden schematisch hinsichtlich des Aufwands je Handover in Abhängigkeit von der Zwischenhandoverzeit verglichen. Der Erwartungswert des Aufwandes läßt sich reduzieren, wenn man für jeden Dienst jeweils die Methode mit dem geringeren zu erwartenden Aufwand – ausgedrückt durch eine Kostenmetrik – auswählt [**TH06**].

5.3.1 Normierte Kostenmetrik

Um den Aufwand bewerten und vergleichen zu können, der mit dem Diensttransfer beziehungsweise der Datentunnelung verbunden ist, wird im folgenden ein kostenbasierter Ansatz vorgeschlagen.

Die Kosten C_{tr} für den *Kontexttransfer* bestehen im Wesentlichen aus zwei Anteilen, nämlich den Kosten für die Bearbeitung des Kontextes einerseits sowie für dessen Übertragung andererseits. Ersteres umfaßt die erforderlichen Prozessortakte, um am vorherigen Aufenthaltsort die Kontextinformation aus dem Dienst zu extrahieren und für die Übertragung vorzubereiten, sowie jene für die Initialisierung des Dienstes am neuen Aufenthaltsort. Die Zahl der Rechenzyklen wird dabei mit der Rechengeschwindigkeit normiert, woraus sich dann die reine Rechendauer ergibt. Die



(a) Kontexttransfer (b) Datentunnelung (c) Ideale Kombination

Abbildung 5.8: Schematische Funktion des Aufwandes je Handover.

Übertragung des Kontextes läßt sich durch seine effektive Übertragungsdauer angeben, also der Datenmenge relativ zur Übertragungsrate.

$$C_{\mathrm{tr}} = \frac{\text{Prozessortakte für den Transfer}}{\text{Prozessorgeschwindigkeit}} + \frac{\text{Kontextgröße}}{\text{Übertragungsrate}} \tag{5.1}$$

Die Kosten für die *Tunnelung* des Datenstroms einer Anwendung zum aktuellen Aufenthaltsort eines mobilen Terminals läßt sich in einen konstanten und einen linearen Anteil aufteilen. Die konstanten Kosten C_{tnl} hängen vom Aufwand für das Einrichten der Tunnelung ab. Da es sich beim Mobilitätsrouting um einen besonders leichtgewichtigen Dienst handelt, ist dieser Aufwand jedoch immer geringer als ein Kontexttransfer. Der Faktor C'_{tnl}, der die linearen Kosten für die Aufrechterhaltung der Tunnelung repräsentiert, ist bestimmt durch die Übertragungsrate des Anwendungsdatenstromes, wie oben normalisiert mit der Datenrate der Leitung.

$$C_{\mathrm{tnl}} = \frac{\text{Prozessortakte für die Tunnelung}}{\text{Prozessorgeschwindigkeit}} + \frac{\text{Signalisierungsdaten}}{\text{Übertragungsrate}} \tag{5.2}$$

$$C'_{\mathrm{tnl}} = \frac{\text{Datenrate der Anwendung}}{\text{Übertragungsrate}} \tag{5.3}$$

Mittels dieser Kostenkoeffizienten läßt sich die *Break-Even-Zeit* t_{b} definieren, zu der die Kosten für einen Diensttransfer und die Datentunnelung gleich groß sind:

$$C_{\mathrm{tr}} = C_{\mathrm{tnl}} + C'_{\mathrm{tnl}} \cdot t_{\mathrm{b}} \tag{5.4}$$

Die Break-Even-Zeit ist eine spezifische Eigenschaft eines jeden Dienstes, die die mit einem Kontexttransfer verbundenen Kosten und die von der Datenrate der Anwendung abhängigen Kosten für eine Tunnelung gegeneinander aufwiegt.

5.3.2 Einheitsstrategie für die Dienstmobilität

Basierend auf dieser Kostenmetrik wird die Mobilitätssteuerung im Falle eines Handover für jeden Dienst eines Teilnehmers separat entscheiden, ob der Dienst transferiert wird oder eine Datentunnelung vorzuziehen ist. Ohne Information über das individuelle Mobilitätsverhalten eines Teilnehmers kann diese Entscheidung lediglich auf Basis der stochastischen Eigenschaften aller Teilnehmer gefällt werden. Diesem Grundsatz folgt die hier vorgestellte *Einheitsstrategie* (ES).

Im folgenden wird gezeigt, daß dann eine hinreichend gute Entscheidung getroffen werden kann, wenn die Kostenkoeffizienten C_{tr} für den Transfer, C_{tnl} für die Einrichtung der Datentunnelung sowie C'_{tnl} für die linearen Kosten der Tunnelung bekannt sind. Aus diesen Koeffizienten läßt sich die Break-Even-Zeit t_{b} einer bestimmten Anwendung – mit einem bekannten Dienst und bekannter Datenrate – berechnen.

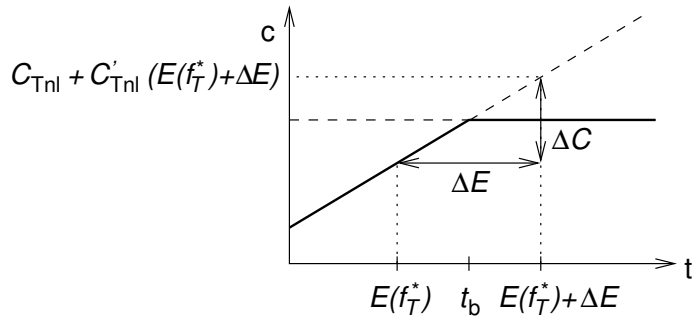


Abbildung 5.9: Fehler der Einheitsstrategie bei falscher Schätzung der Zwischenhandoverzeiten.

Die Wahrscheinlichkeitsdichte $f_T^\star(t)$ und der Erwartungswert $E(f_T^\star)$ der Zwischenhandoverzeiten wurden in Abschnitt 3.3.2 hergeleitet[2]. Eine Entscheidung auf Grundlage der stochastischen Eigenschaften aller Teilnehmer ist gleichbedeutend mit dem Vergleich von t_b und dem Erwartungswert $E(f_T^\star)$. Ist $E(f_T^\star)$ größer als t_b, so wird ein Kontexttransfer ausgeführt. Andernfalls wird die Datenweiterleitung initiiert. Der aus der Einheitsstrategie resultierende Erwartungswert der Kosten je Handover ist somit

$$E_C^{\mathrm{ES}} = \min\left(C_\mathrm{tnl} + C'_\mathrm{tnl} \cdot E(f_T^\star),\ C_\mathrm{tr}\right) \tag{5.5}$$

Liegt der angenommene Erwartungswert des Handoverintervalls durch einen systematischen Fehler um ein gewisses ΔE neben dem wahren Erwartungswert $E(f_T^\star)$, so macht sich dies nur dann bemerkbar, wenn daraufhin auch tatsächlich eine *Fehlentscheidung* beispielsweise für einen Transfer getroffen wird, wenn eigentlich eine Tunnelung weniger Aufwand hätte erwarten lassen, also

$$\mathrm{sgn}(E(f_T^\star) - t_\mathrm{b}) = -\mathrm{sgn}(E(f_T^\star) + \Delta E - t_\mathrm{b}) \tag{5.6}$$

ist. Der Erwartungswert der Kosten je Handover erhöht sich dadurch um

$$\Delta C = C'_\mathrm{Tnl} \cdot |\Delta E| \tag{5.7}$$

wie in Abb. 5.9 für den Fall aufgetragen, daß statt einem Transfer fälschlicherweise für die Datentunnelung entschieden wird. Ist umgekehrt der tatsächliche Erwartungswert der Zwischenhandoverzeiten größer als die Break-Even-Zeit und der fehlerbehaftete Schätzwert kleiner – so daß irrtümlicherweise ein Diensttransfer eingeleitet wird – gilt aufgrund der Symmetrie der gleiche Zusammenhang.

Darüber hinaus ließe eine individuelle Prädiktion der Aufenthaltsdauer – anstatt des Erwartungswertes aller Teilnehmer – eine weitere Senkung der Kosten je Handover erwarten. Um den dafür vorhandenen Spielraum zu ermitteln, soll nun der optimale Fall mit *vollkommener Prädiktion* eines jeden Handover berechnet werden.

[2]Die Wahrscheinlichkeitsdichtefunktion $f_T^\star(t)$ beschreibt die Verteilung der Zwischenhandoverzeiten unter Berücksichtigung des Umstandes, daß schnellere Teilnehmer häufigeren Handovern unterliegen.

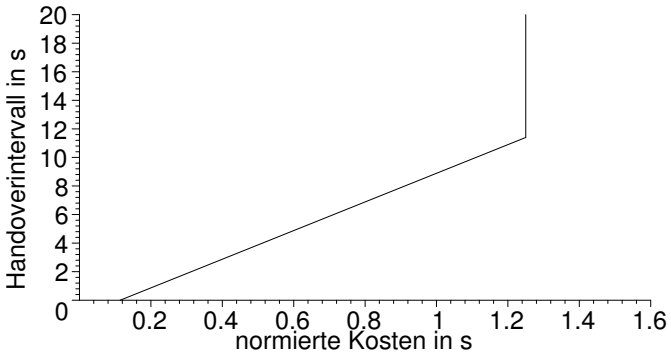


Abbildung 5.10: Inverse Kostenfunktion $t(c)$, vgl. Abb. 5.8(c).

5.3.3 Theoretisches Kostenoptimum der Dienstmobilität

Für eine optimale Entscheidung zwischen Diensttransfer und Datentunnelung müsste die genaue Zeit bis zum nächsten Handover bekannt sein. Für "schnelle" Teilnehmer mit einer kurzen Zeit bis zum nächsten Handover (kürzer als t_b) würde dann die Datentunnelung verwendet, während hingegen bei "langsamen" Teilnehmern (Zwischenhandoverzeit länger als t_b) ein Kontexttransfer durchgeführt würde. Aus der Wahrscheinlichkeitsdichtefunktion der Zwischenhandoverzeiten $f_T^\star(t)$ und der in Abb. 5.10 gezeigten inversen Kostenfunktion $t(c)$ läßt sich entsprechend der Parametertransformation

$$f_c(c) = f_T^\star(t(c)) \left| \frac{\partial t(c)}{\partial c} \right| \tag{5.8}$$

mit $c(t) = \min\left(C_\mathrm{tnl} + C'_\mathrm{tnl} \cdot t,\ C_\mathrm{tr}\right)$ die Wahrscheinlichkeitsdichtefunktion der Kosten im Idealfall mit vollkommener Bewegungsprädiktion herleiten als

$$f_c(c) = f_T^\star \left(\frac{c - C_\mathrm{tnl}}{C'_\mathrm{tnl}} \right) \cdot \frac{1}{C'_\mathrm{tnl}} + \delta(C_\mathrm{tr}) \cdot \int_{t_\mathrm{b}}^{\infty} f_T^\star(t)\,\mathrm{d}t \tag{5.9}$$

für $C_\mathrm{tnl} < c < C_\mathrm{tr}$. Man beachte, daß der konstante Abschnitt der Kostenfunktion einen Dirac-Impuls bei C_tr ergibt, wie in Abb. 5.11 ersichtlich. Für den Graphen wurden die Werte $C_\mathrm{tr} = 0{,}11 \cdot t_\mathrm{b}$, $C_\mathrm{tnl} = 0{,}01 \cdot t_\mathrm{b}$ und $C'_\mathrm{tnl} = 0{,}1$ gewählt.

Für den Erwartungswert der Kosten im optimalen Fall ergibt sich daraus

$$E_C^\mathrm{Opt}(t_\mathrm{b}) = \int_0^{t_\mathrm{b}} (C_\mathrm{tnl} + C'_\mathrm{tnl} \cdot t)\, f_T^\star(t)\,\mathrm{d}t + C_\mathrm{tr} \cdot \int_{t_\mathrm{b}}^{\infty} f_T^\star(t)\,\mathrm{d}t \tag{5.10}$$

wobei man deutlich die beiden Anteile für die Tunnelung (für Zwischenhandoverzeiten von 0 bis t_b) und den Transfer (t_b bis ∞) erkennen kann. Die Erwartungswerte der Kosten der Einheitsstrategie und das theoretische Optimum sind in Abb. 5.12 in Abhängigkeit von der Break-Even-Zeit t_b aufgetragen. Dabei wurden die konstanten Werte $C_\mathrm{tr} = 0{,}11 \cdot E(f_T^\star)$, $C_\mathrm{tnl} = 0{,}01 \cdot E(f_T^\star)$ verwendet und $C'_\mathrm{tnl}(t_\mathrm{b}) = \frac{C_\mathrm{tr} - C_\mathrm{tnl}}{t_\mathrm{b}}$ variabel gewählt. Diese Betrachtung zeigt die Wirkung der

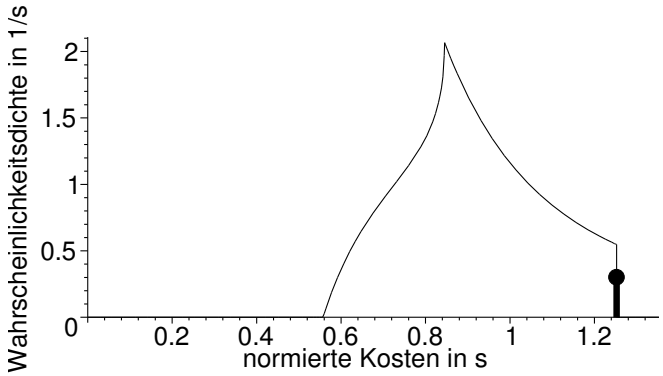


Abbildung 5.11: Wahrscheinlichkeitsdichtefunktion der Kosten je Handover im Idealfall.

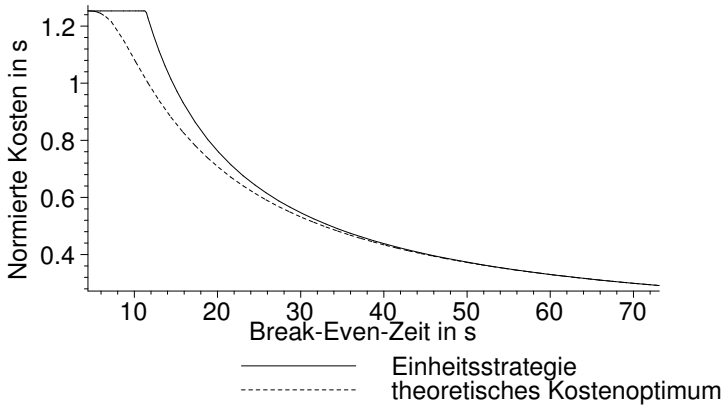


Abbildung 5.12: Erwartungswert der Kosten je Handover in Abhängigkeit von der Break-Even-Zeit t_b (durch Variation von C'_tnl).

Einheitsstrategie im Falle der Verwendung eines bestimmten Dienstes mit unterschiedlichen Datenraten. Erwartungsgemäß liegt das theoretische Kostenoptimum stets unterhalb den Kosten der Einheitsstrategie. Es ist nicht überraschend, daß der Unterschied zwischen der Einheitsstrategie und dem optimalen Fall dort am größten ist, wo der Wert von t_b gleich dem Erwartungswert der Zwischenhandoverzeit des mobilen Teilnehmers ist, da genau dann die Zahl von Fehlentscheidungen der Einheitsstrategie am größten ist.

5.3.4 Suboptimum durch die Ungenauigkeit der Bewegungsschätzung

Bedingt durch das Prinzip der Kausalität läßt sich die Zwischenhandoverzeit nie exakt vorhersagen. Sie läßt sich jedoch beispielsweise aus der Bewegung des Teilnehmers in der Vergangenheit oder an Hand anderer Kriterien mit einer gewissen Genauigkeit schätzen. Je besser diese Schätzung ausfällt, umso näher reichen die tatsächlichen Kosten an das theoretische Optimum heran. Für die nachfolgenden Betrachtungen wird angenommen, daß die Schätzungen vom wahren Wert

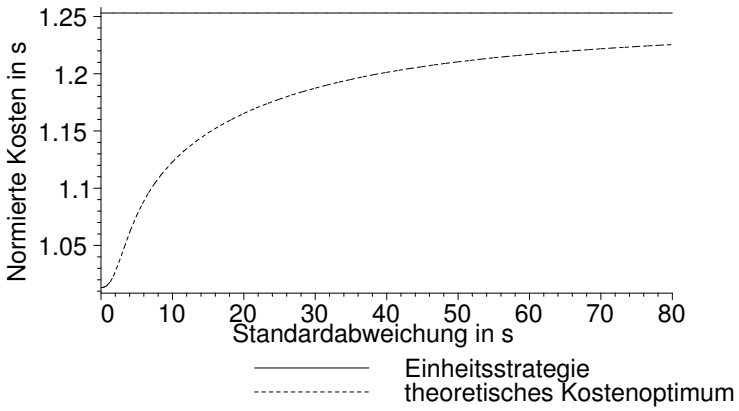


Abbildung 5.13: Erwartungswert der Kosten je Handover, wobei die Break-Even-Zeit t_b gleich dem Erwartungswert der Zwischenhandoverzeit ist.

entsprechend einer Gauß'schen Normalverteilung mit Standardabweichung σ abweichen. Führt die Abweichung vom wahren Wert zu einer Fehlentscheidung – Tunnelung statt Kontexttransfer oder umgekehrt – fallen höhere Kosten an als erforderlich. Je größer die Standardabweichung σ, um so höher ist die Wahrscheinlichkeit einer falschen Entscheidung und um so höher sind auch die resultierenden Kosten.

Da die für diesen Fall geltende Kostenfunktion $c(t)$ sich nicht invertieren läßt, kann der Erwartungswert der Kosten in Abhängigkeit von der Standardabweichung σ der Bewegungsprädiktion lediglich aufwendiger angegeben werden als

$$E_C(\sigma) = \int_0^\infty f_T^\star(t) \left[(C_\mathrm{tnl} + C'_\mathrm{tnl} \cdot t)\, P_\mathrm{tnl} + C_\mathrm{tr} \cdot\ P_\mathrm{tr} \right] \mathrm{d}t \tag{5.11}$$

wobei P_tnl und P_tr (die Wahrscheinlichkeit daß für eine Tunnelung beziehungsweise für einen Transfer entschieden wird) Funktionen von t_b, t und σ sind:

$$P_\mathrm{tnl} = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{t_\mathrm{b}} \mathrm{e}^{-\frac{(\tau-t)^2}{2\sigma^2}}\, \mathrm{d}\tau \tag{5.12}$$

$$P_\mathrm{tr} = \frac{1}{\sigma\sqrt{2\pi}} \int_{t_\mathrm{b}}^{\infty} \mathrm{e}^{-\frac{(\tau-t)^2}{2\sigma^2}}\, \mathrm{d}\tau \tag{5.13}$$

Die Erwartungswerte der Kosten sind in den Abb. 5.13 und 5.14 in Abhängigkeit von der Standardabweichung der Bewegungsprädiktion aufgetragen. Dabei wurden in Abb. 5.13 die Kostenkoeffizienten des Kontexttransfers sowie die Datenrate der Anwendung so angenommen, daß sich eine Break-Even-Zeit ergibt, die gleich dem Erwartungswert der Zwischenhandoverzeit ist. Man erkennt, daß in diesem besonderen Fall jedwede – selbst äußerst ungenaue – Information dazu beiträgt, den Erwartungswert der Kosten zu senken. Für $\sigma \to \infty$ nähert sich das theoretisch erreichbare Kostenoptimum der Einheitsstrategie an.

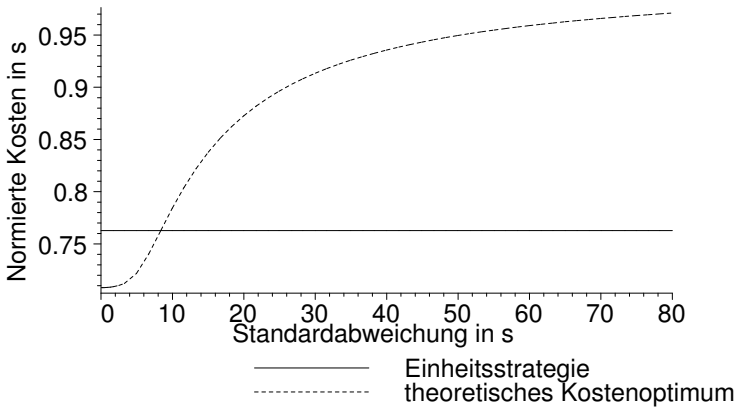


Abbildung 5.14: Erwartungswert der Kosten je Handover mit einer Break-Even-Zeit $t_\mathrm{b} = 20\,\mathrm{s}$.

In Abb. 5.14 wurden hingegen Parameter gewählt, die eine Break-Even-Zeit von 20 s ergeben. Bei den hier angenommenen Bewegungsparametern wählen dann 90% der Terminals die Datenweiterleitung, während die restlichen 10% einen Kontexttransfer ausführen. Wie man sieht, tragen nur vergleichsweise präzise Schätzwerte zur Verbesserung gegenüber der Einheitsstrategie bei. Die beiden Kurven schneiden sich bei einer Standardabweichung von etwa 8 s, was folglich die minimale erforderliche Standardabweichung ist, um unter diesen Umständen aus einer Bewegungsvorhersage einen Vorteil zu erlangen.

In Abb. 5.15 sind diese minimal erforderlichen Standardabweichungen in Abhängigkeit von der Break-Even-Zeit aufgetragen. Ist t_b gleich dem Erwartungswert der Zwischenhandoverzeit, geht die minimal erforderliche Standardabweichung gegen Unendlich, wie bereits oben dargelegt (siehe auch Abb. 5.13).

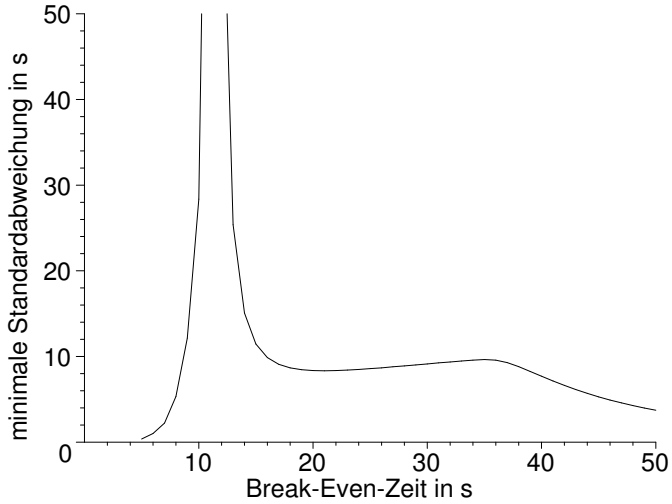


Abbildung 5.15: Erforderliche minimale Standardabweichung für eine Verbesserung gegenüber der Einheitsstrategie.

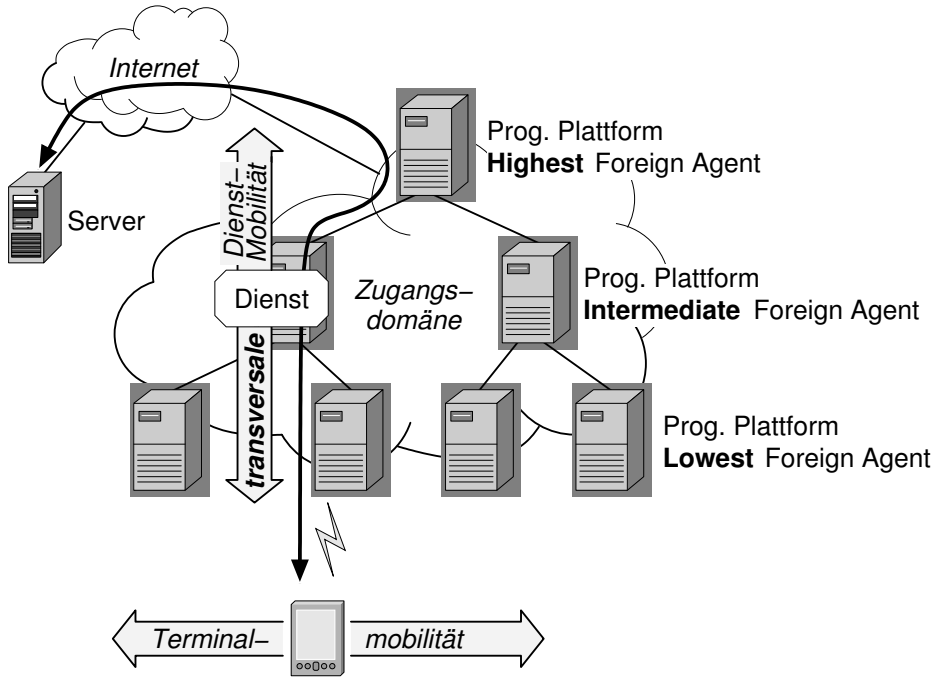


Abbildung 5.16: Transversale Dienstmobilität im Zugangsbereich.

5.4 Transversale Dienstmobilität

Eine andere Möglichkeit, den Aufwand je Handover zu reduzieren, eröffnet sich, indem man bei *Hierarchischem Mobile IP* (siehe Abschnitt 2.2.3) die Hierarchie der Foreign Agents in die Standortsuche für den mobilen Dienst einbezieht. Hierarchisches Mobile IP verringert den globalen Signalisierungsaufwand, indem eine Hierarchie von Foreign Agents in der Zugangsdomäne eingerichtet wird. Bei einem Handover innerhalb der logischen Hierarchie einer Domäne endet dabei die Signalisierung bei demjenigen Foreign Agent innerhalb der Hierarchie, der sowohl für den vorherigen als auch den aktuellen Aufenthaltsort des Teilnehmers zuständig ist. Erst dann, wenn das mobile Terminal bei einem Netzwechsel zugleich die Domäne seines Highest Foreign Agent verläßt, wird der Home Agent informiert.

Daraus resultiert, daß der Routingpfad aller Pakete eines Teilnehmers stets über den Highest Foreign Agent führt, solange sich der Teilnehmer innerhalb dieser Domäne bewegt. Verlagert man nun den Dienst eines mobilen Teilnehmers vom Foreign Agent vor Ort auf den Highest Foreign Agent, so kann der Dienst auch an seinem Ort belassen werden, solange kein Handover in den Bereich eines anderen Highest Foreign Agent erfolgt. Daraus ergibt sich ein zusätzlicher Freiheitsgrad für mobile programmierbare Dienste, da diese sich somit auch transversal zur Bewegung des mobilen Terminals bewegen können, wie in Abb. 5.16 dargestellt [**TS03**].

Aus diesem Ansatz resultiert nun eine Abwägung zwischen einer Reduktion der Zahl der Diensttransfere und den begrenzten Ressourcen des Highest Foreign Agent. Unter Verwendung der in [Bic05] bestimmten Metriken läßt sich diese Entscheidung auf den Fall der in Abschnitt 2.1.3

vorgestellten Overlaynetze bei GSM zurückführen. Ein intelligentes Management wird folglich diejenigen Dienste auf den Highest Foreign Agent verlagern, die den höchsten Transferaufwand erwarten lassen. Im Gegenzug werden solche Dienste, die einen geringen Transferaufwand erwarten lassen (da entweder der Aufwand je Handover gering ist oder der zugehörige Teilnehmer sich langsam bewegt) auf die in der Hierarchie niederen Foreign Agents verwiesen.

5.5 Zusammenfassung

In diesem Kapitel wird zunächst auf die Mobilität von Diensten eingegangen. Dabei werden der *Kontexttransfer* und der *Diensttransfer* voneinander abgegrenzt und dargelegt, weshalb der Kontexttransfer lediglich eine Untermenge eines Diensttransfers darstellt. Es wird dann analysiert, weshalb es vorteilhaft ist die programmierbare Plattform, auf der sowohl die Dienste als auch die zugehörige Steuerung dynamisch geladen werden, mit dem Foreign Agent zu kollokieren.

Darauf aufbauend wird eine Architektur entworfen, die *programmierbare Dienste* für *mobile Teilnehmer* bereitstellt. Diese ist modular aufgebaut und kann daher ihrerseits von der Eigenschaft der Programmierbarkeit Nutzen ziehen. Charakterisierend für die Architektur ist einerseits eine vollständige Abspaltung der mobilitätsbezogenen Elemente sowie die Fähigkeit, einen *Diensttransfer* oder eine *Datentunnelung* zur Bereitstellung von Diensten für mobile Teilnehmer auszuführen. Ein besonderes Merkmal ist dabei, daß dies selektiv je Anwendungsdatenstrom beziehungsweise Dienst geschehen kann.

Welche der beiden Methoden den geringeren Aufwand erwarten läßt, hängt dabei sowohl von der erwarteten Zwischenhandoverzeit als auch den Handovereigenschaften des programmierbaren Dienstes ab, die durch die *Break-Even-Zeit* charakterisiert sind. Für die Entscheidung, wann ein Diensttransfer oder eine Datentunnelung einzuleiten ist, wird eine heuristische Einheitsstrategie vorgeschlagen. Es wird gezeigt, daß es vorteilhafter ist, diese Entscheidung dynamisch je Dienst und Anwendungsdatenstrom zu fällen, anstatt für alle Dienste die Verwendung einer bestimmten der beiden Variante vorzugeben. Schließlich wird die Einheitsstrategie mit dem theoretischen Optimum verglichen. Dabei wird festgestellt, daß der Erwartungswert der Kosten bei Verwendung der Einheitsstrategie nicht wesentlich über dem theoretischen Minimum liegt, so daß aufwendigere Strategien in diesem Punkt keine wesentliche Verbesserung erreichen können.

Kapitel 6

Realisierung eines transparenten Programmierbaren Proxy für mobile Terminals

Anwendungen mobiler Teilnehmer, die sich über Netzgrenzen hinwegbewegen, hinsichtlich der Kommunikation mit entfernten Servern optimal zu unterstützen, ist der Zweck des *Programmierbaren Proxy*. Dieser fungiert als intelligenter Stellvertreter des mobilen Teilnehmers im Festnetz und sorgt dafür, daß die vom Teilnehmer angeforderten Daten im Hinblick auf Durchsatz und Latenz bestmöglichst *angefordert*, *aufbereitet* und diesem *zugestellt* werden [**TBL03**]. Die Nutzung des Programmierbaren Proxy sollte dabei für die entsprechenden Anwendungen transparent sein, also keine Änderungen der Anwendungs- oder Protokoll-Logik erfordern.

6.1 Einsatzszenario

Für die prototypische Untersuchung des *Programmierbaren Proxy* wird von einem mobilen Terminal ausgegangen, welches mit einem entfernten WWW-Server kommuniziert und sich währenddessen – mit Mobile IP zur Mobilitätsunterstützung – zwischen verschiedenen IP-Netzen bewegt.

Verwendet ein Terminal zur Mobilitätsunterstützung Mobile IP, kann mittels des Programmierbaren Proxy die Anforderung der Daten direkt erfolgen und so der Umweg über den Home Agent vermieden werden[1]. Dies erhöht in der Regel den maximal erreichbaren Datendurchsatz, da die Flußkontrolle bei Fensterprotokollen (z. B. TCP, SCTP) von der Paketumlaufdauer abhängt.

[1] Dies setzt jedoch voraus, daß der Proxy für die Anfrage seine eigene IP-Adresse verwendet. Eine etwaige Authentifizierung anhand der IP-Adresse des Teilnehmers bei der Gegenstelle wird daher nicht funktionieren. Da dieses Problem jedoch allgemein bei jeder Art von Proxy besteht, wird eine Authentifizierung mittels der IP-Adresse gemeinhin abgelehnt [CD01].

Eine *Aufbereitung* der Daten für den mobilen Teilnehmer ist dann erforderlich, wenn sein Gerät nicht über die für bestimmte Anwendungen notwendigen Ressourcen verfügt. Dies können beispielsweise die Größe der Bedienanzeige, Art und Umfang der Eingabemöglichkeiten oder die verfügbare Rechenleistung sein.

Die *Zustellung* vom Proxy zum mobilen Teilnehmer erfolgt – wie auch bei Mobile IP – unter Umgehung des regulären Routings direkt, solange sich der Teilnehmer im lokalen Netz aufhält. Hat er sich allerdings seit dem Absenden der Anfrage in ein anderes Netz bewegt, müssen die Daten zum neuen Aufenthaltsort weitergeleitet werden. Die Performanz der Zustellung an den Teilnehmer kann verbessert werden, indem anstatt des TCP ein Transportprotokoll zum Einsatz kommt, dessen Flußkontrolle nach Fertigstellung des Handover die Datenübertragung sofort wieder aufnimmt.

Bei zustandsbehafteten Protokollen ist es erforderlich, die Datentunnelung weiterhin über den gleichen Proxy zu führen, der die ursprüngliche Anfrage des Terminals zur Initiierung der Sitzung angenommen hat. Dies gilt selbstverständlich für alle laufenden TCP-Verbindungen, da diese keinen Wechsel der IP-Adresse dulden. Der Begriff *zustandsbehaftet* ist im Falle mobiler Teilnehmer jedoch nur bezüglich der tatsächlichen Bindung eines Zustandes an eine bestimmte IP-Adresse zu verstehen. Das HTTP ist beispielsweise ein an sich *zustandsloses* Protokoll, welches das zustandsbehaftete TCP zum Transport nutzt. Trotz dieser darunterliegenden Zustände kann daher eine HTTP-Anfrage nach einem Handover auch vom neuen Proxy bearbeitet werden.

6.2 Konzept des Programmierbaren Proxy

Im Vordergrund des realisierten Prototyps stand die Mobilität und insbesondere die Einbettung des *Programmierbaren Proxy* in die mobile Umgebung, so daß der mobile Teilnehmer diesen transparent über Netzwechsel hinweg nutzen kann. Der Proxy sendet dann die ihm übergebenen HTTP-Anfragen direkt an den Zielserver und nicht zuerst durch den IP-Tunnel zum Home Agent des mobilen Terminals.

Dies entspricht vom Grundgedanken her den in [ZCB01] vorgeschlagenen *Mobile Policy Tables*. Bei diesem Ansatz verwendet das Terminal die topologisch korrekte IP-Adresse (anstatt der unveränderlichen IP-Adresse aus dem Heimatnetz) und vermeidet dabei den Umweg über den Home Agent. Da jedoch der Endpunkt der Transportverbindung im Terminal angesiedelt ist, brechen diese Transportverbindungen im Falle eines Handover ab.

Sowohl die Proxy-basierte Lösung wie auch die Mobile Policy Tables erzielen im Prinzip den gleichen Effekt wie die in [PJ98, JPA04] beschriebene Routenoptimierung[2] für Mobile IP. Allerdings ist letztere ausschließlich auf der Vermittlungsschicht angesiedelt und zieht Informationen höherer Schichten nicht in Betracht. Dies schränkt den Nutzen dieser Lösung zusätzlich ein. Verwendet der mobile Teilnehmer unnötigerweise einen HTTP-Proxy in seinem Heimatnetz, wird

[2]Obwohl von der IETF propagiert, hat die Routenoptimierung insbesondere aufgrund der in Abschnitt 2.2.4 genannten Probleme bislang keinerlei Verbreitung gefunden.

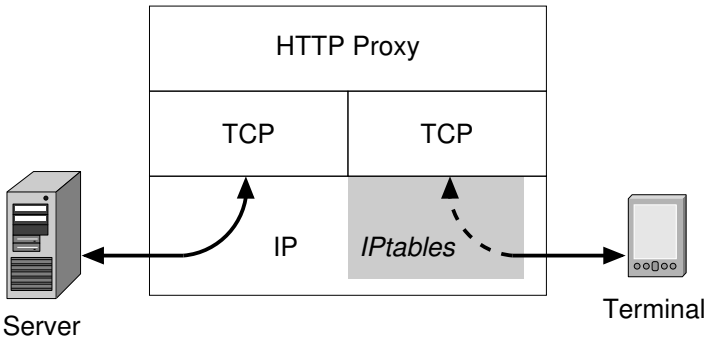


Abbildung 6.1: Transportverbindungen bei einem transparenten HTTP-Proxy.

die Transportschicht eine Verbindung zwischen dem Endgerät und dem Proxy im Heimatnetz herstellen, weshalb eine auf die IP-Schicht beschränkte Routenoptimierung dann keinerlei Vorteil bringt.

6.2.1 Selektiver Transparenter Proxy

Ein *transparenter HTTP-Proxy* vereinigt die Funktion eines regulären HTTP-Proxy mit der Eigenschaft, daß er IP-Pakete entgegennehmen und verarbeiten kann, die nicht seine IP-Adresse als Empfängeradresse tragen. Dazu müssen diese Pakete wie in Abb. 6.1 gezeigt aus der regulären Paketverarbeitung herausgenommen, umadressiert und zum Proxy umgelenkt werden. Dies geschieht beispielsweise mittels des Paketverarbeitungssystems *iptables* des Projektes netfilter.org [Wel05] mit dem Aufruf:

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080
```

Dieser bewirkt, daß alle beim Rechner eingehenden IP-Pakete unmittelbar nach ihrem Eintreffen (`-A PREROUTING`) dahingehend untersucht werden, ob das Protokoll TCP (`-p tcp`) und zugleich Port 80 das Ziel ist (`--dport 80`). Ist beides der Fall, so werden die Zieladresse auf den lokalen Rechner (`-j REDIRECT`) und der Zielport auf den Proxy-Port 8080 (`--to-ports 8080`) geändert. Da der Eintrag in der NAT-Tabelle erfolgt ist (`-t nat`), findet für die vom Proxy zurückgesandten Pakete eine entsprechende Adressrückumsetzung statt.

Auf diese Weise hat der Teilnehmer den Eindruck, er kommuniziere unmittelbar mit dem entfernten Server. Tatsächlich besteht die Transportverbindung jedoch mit dem Proxy, der seinerseits eine Transportverbindung zum Zielserver aufbaut. Dazu müssen jedoch alle von dem betreffenden Teilnehmer gesendeten Pakete auch zuverlässig durch den Proxy geleitetet werden. Umgeht nämlich eines der Datagramme den Proxy und erreicht den Server direkt, wird dieser mit einer Fehlermeldung (TCP Reset) antworten, woraufhin das Terminal dann seine Transportverbindung mit einer Fehlermeldung an die entsprechende Anwendung schließen wird.

Im stationären Fall nimmt der Programmierbare Proxy die HTTP-Anfrage transparent entgegen und sendet diese – unter Verwendung seiner topologisch korrekten IP-Adresse – an der Mobile IP-Infrastruktur vorbei direkt an den entsprechenden WWW-Server, wie in Abb. 6.2 dargestellt.

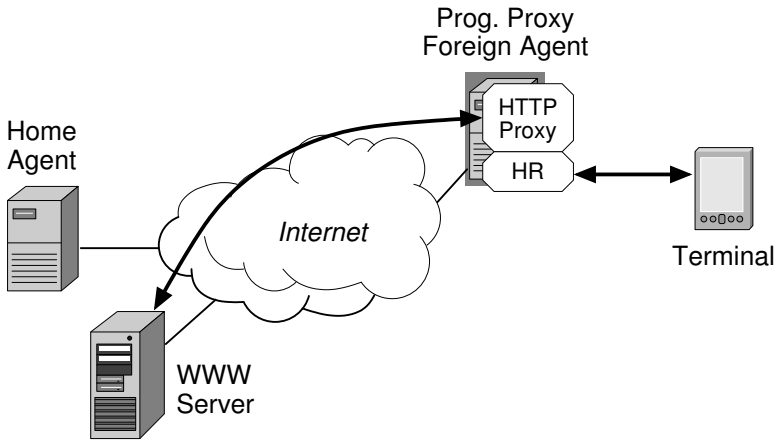


Abbildung 6.2: Höheres Routing (HR) für den Programmierbaren Proxy.

Die Antwort des Servers trifft über die gleiche Transportverbindung wieder beim Proxy ein und wird dann – wie oben beschrieben – nach Adressumsetzung dem anfragenden Terminal zugestellt.

In den Fällen, in denen der Proxy auf einem Gateway angeordnet wird oder die Pakete vom Teilnehmer selbst IP-in-IP-enkapsuliert und dekapsuliert werden, treffen die Segmente der HTTP-Verbindung IP-in-IP-enkapsuliert beim Proxy ein. Um zu entscheiden, welche Pakete dann an den lokalen Proxy umgeleitet werden, muß folglich das *höhere Routing* (HR) den TCP-Header im inneren IP-Paket finden und analysieren. Bei der Umleitung an den lokalen Proxy muß zuerst die Enkapsulierung entfernt werden, bevor die Umadressierung erfolgt. Umgekehrt muß auch bei den für den mobilen Teilnehmer bestimmten Paketen die Adresse rückübersetzt und jedes Paket wieder IP-in-IP-enkapsuliert werden, um es in den Tunnel einzufügen. Das Ergebnis ist ein selektiver Tunnelendpunkt für z. B. HTTP-Ströme, der mit anderen Arten von Daten jedoch nicht interferiert.

Im Schichtendiagramm in Abb. 6.3 ist das Zusammenspiel der einzelnen Einheiten hinsichtlich der Nutzdatenströme gezeigt. Auf der Vermittlungsschicht befindet sich sowohl im Terminal als auch im programmierbaren Knoten das Mobile IP, welches wieder das IP verwendet – angedeutet durch die Verzahnung. Im Foreign Agent verwendet Mobile IP direkt Funktionen der Sicherungsschicht, um Datagramme an den Teilnehmer zuzustellen. Ebenfalls auf der Vermittlungsschicht, von der Funktionalität her jedoch auch unterhalb von IP, sind die bereits erwähnten IPtables angesiedelt. Die hier verwendete programmierbare Plattform *AMnet* [FHSZ02] setzt direkt auf der Vermittlungsschicht auf und verwendet dabei auch die Funktionen von IPtables. Sie kann sowohl direkt auf IP-Pakete zugreifen oder auch TCP und UDP verwenden, daher ist hier eine Verzahnung vorhanden. In der programmierbaren Plattform lassen sich dynamisch Module laden, die Funktionen bis hin zur Anwendungsschicht beinhalten können.

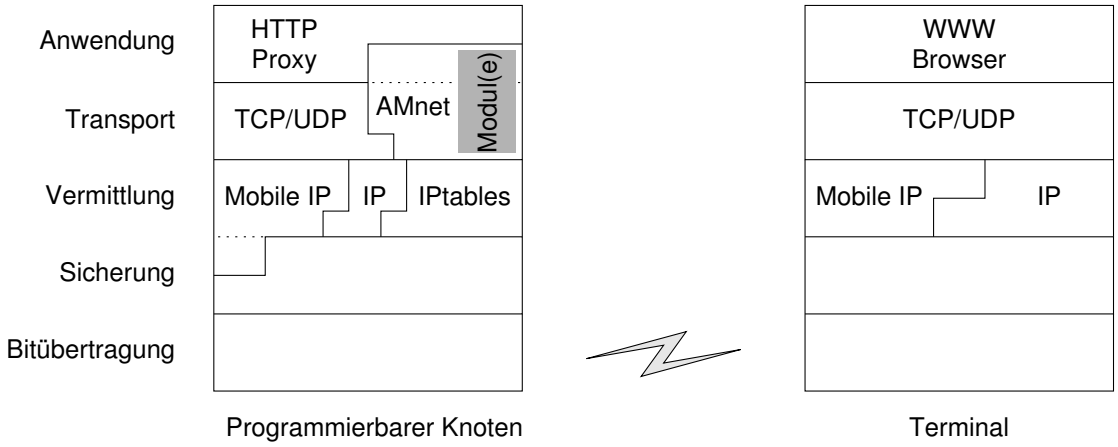


Abbildung 6.3: Softwareschichtung des Programmierbaren Proxy.

6.2.2 Unterstützung von Netzwechseln

Wechselt der Teilnehmer sein IP-Netz und somit in den Bereich eines anderen Foreign Agent, können neue Transportverbindungen vom Proxy an diesem neuen Ort angenommen und bedient werden. Bestehende Verbindungen wurden jedoch mit der topologisch korrekten IP-Adresse des vorherigen Proxy initiiert. Da Datagramme dieser Verbindungen also weiterhin in das vorherige IP-Netz gelenkt werden, ergäbe ein Transfer des Proxy-Dienstes keinerlei Vorteil. Daher ist in diesem Falle die *Tunnelung* der Verbindungen vom beziehungsweise zum neuen Aufenthaltsort des mobilen Teilnehmers vorteilhaft. Dabei müssen Pakete bestehender Verbindungen vom höheren Routing stets zu demjenigen Proxy geleitet werden, der diese Verbindung ursprünglich angenommen hat. Das höhere Routing führt dabei also die in Abschnitt 5.2 als *Mobilitätsrouting* auf der Netzschicht eingeführte Funktionalität anhand von Merkmalen höherer Schichten durch, wie in Abb. 6.4 dargestellt.

Um die Pakete bestehender Verbindungen dem jeweils zuständigen vorherigen Proxy zuschicken zu können, benötigt der aktuelle Proxy eine Tabelle mit den entsprechenden Zuordnungen. Diese wird als Teil des Dienstkontextes bei jedem Handover an den jeweils neuen Proxy weitergereicht. Nach einem Handover darf der neue Programmierbare Proxy jedoch erst dann Pakete vom mobilen Terminal annehmen (oder von der Bearbeitung ausnehmen und in Richtung Home Agent durchlassen), wenn feststeht, daß diese nicht zu einer bereits bestehenden Verbindung gehören, die von einem vorherigen Proxy bearbeitet wird[3]. Im Fall von verbindungsorientierten Protokollen – wie insbesondere dem TCP – vereinfacht sich diese strikte Vorgabe, da ein leicht erkennbarer Verbindungsanfang existiert. Bei anderen Protokollen ergibt sich jedoch ein Zielkonflikt zwischen einer Performanzverbesserung im statischen Fall und einer möglichen verlängerten Handoverlatenz.

[3] Sonst besteht die Gefahr eines Abbruchs der Verbindung, siehe oben.

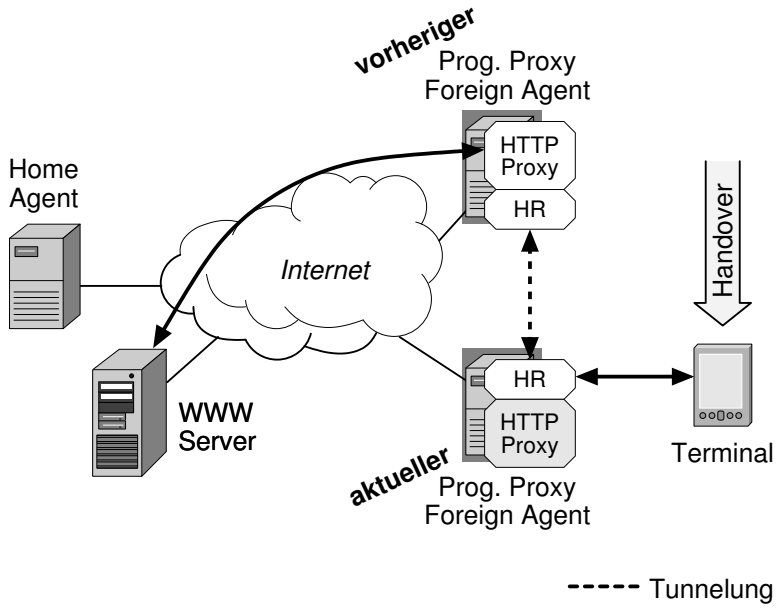


Abbildung 6.4: Höheres Routing (HR) zur Handoverunterstützung laufender Verbindungen.

6.2.3 Handover-Triggered TCP (hot-TCP)

Bei einem *break-before-make* Handover lassen sich Verluste von Datagrammen nicht vermeiden, wie in 2.2.1 dargelegt. Diese interpretiert jedoch die Flußkontrolle der meisten Transportprotokolle – wie z. B. die des TCP – als ein Indiz für eine Netzüberlastung und verringert infolgedessen die Sendedatenrate. Das TCP reduziert bei längeren Unterbrechungen der Konnektivität sein Sendefenster bis auf ein einziges Segment und versucht in zunehmend längeren Intervallen, dieses eine Segment an die Gegenstelle zu übermitteln. Der dafür verwendete *Retransmission Timer* verdoppelt sich bei jedem fehlgeschlagenen Versuch entsprechend dem *Back-off*-Algorithmus [PA00]. Dies verursacht den größten Anteil am Performanzverlust bei einem Handover [SS02]. Nach Fertigstellung des Handover kann es daher also noch etliche Sekunden dauern, bis das TCP den nächsten Sendeversuch unternimmt [**DST04**].

Um zu bewirken, daß das TCP die Datenübertragung sofort nach Ende des Handover wiederaufnimmt, muß die Flußkontrolle über die Fertigstellung des Handover unterrichtet werden. Sie kann dann den *Retransmission Timer* zurücksetzen und die nächste Sendewiederholung unverzüglich vornehmen. Erreicht den Sender eine Bestätigung vom Empfänger, steigert der Sender seine Sendedatenrate entsprechend dem *Slow Start*-Algorithmus. Der Name *Slow Start* ist jedoch irreführend – vielmehr erfolgt ein exponentielles Anheben der Größe des TCP-Sendefensters und somit der Sendedatenrate. Bei einer kurzen Paketumlaufzeit zwischen den Endpunkten der Verbindung, wie dies hier der Fall ist, geht der sogenannte *Slow Start* tatsächlich sehr schnell vonstatten.

Bewegt sich der mobile Teilnehmer rasch durch mehrere Netze hindurch, bleibt ihm nach Fertigstellung des Handover bis zum nächsten Handover nur wenig Zeit zur Datenübertragung. In

diesem Fall kann die Verringerung der Datenrate einen merklichen Effekt auf den Datendurchsatz haben [Jac88]. Beim *Handover-Triggered TCP* (hot-TCP) [**TBW04**] wird daher der *Slow Start* nach einem Handover umgangen und beim Zurücksetzen des *Retransmission Timers* zusätzlich das Sendefenster auf den letzten Wert vor der – durch den Handover bedingten – Serie von Paketverlusten zurückgestellt.

Die Weitergabe des Handovertriggers von der Mobilitätssteuerung an die Flußkontrolle kann bei der hier gewählten Architektur einfach realisiert werden, da der netzseitige Endpunkt der Transportverbindung vor Ort im Einflußbereich der Mobilitätssteuerung liegt. Im Gegensatz dazu ist der Endpunkt der Transportverbindung bei Mechanismen auf der Netzschicht – also Mobile IP und seinen Verbesserungen – im entfernten Server beheimatet und entzieht sich daher dem direkten Einfluß.

6.3 Prototypische Implementierung

Mittels einer prototypischen Implementierung soll im folgenden das Handoververhalten des mobilen Netzdienstes untersucht werden. Dazu wird ein minimaler Satz an Funktionen implementiert, der die wesentlichen Aspekte des *kritischen Pfades* nachbildet. Vereinfachend wird angenommen, daß der mobile Teilnehmer sogenannte *Ping-Pong*-Handover zwischen zwei bestimmten IP-Netzen ausführt. Dadurch vereinfacht sich die Realisierung der Mobilitätssteuerung deutlich, da beispielsweise die IP-Adressen der beiden Foreign Agents vorab fest eingestellt werden können. Auch ist eine dynamische Ladbarkeit des Dienstes nicht erforderlich. Da der Dienst manuell konfiguriert und gestartet wird, kann folglich ebenso auf ein Dienstmanagement verzichtet werden.

6.3.1 Struktur des Prototyps

In Abb. 6.5 sind die einzelnen Softwareeinheiten der prototypischen Implementierung mitsamt ihren Kommunikationsbeziehungen dargelegt. Der Nutzdatenstrom ist durch eine dickere Linie gekennzeichnet. Die drei Prozesse, die eigens für diesen Prototyp entwickelt wurden, sind in weiß hervorgehoben.

Die Basis für die Mobilitätsunterstützung – insbesondere hinsichtlich des Aspekts der Mobilitätserkennung – bildet Mobile IP, wobei hier die Implementierung *Dynamics* der Universität Helsinki [FMMW99] zum Einsatz kommt. Mobile IP wird im Foreign Agent-Modus betrieben, so daß sich der mobile Teilnehmer mittels eines Foreign Agent beim Home Agent registriert und der IP-in-IP-Tunnel nur zwischen diesen beiden Einheiten aufgebaut wird – und nicht bis hin zum Terminal. Verbesserungen von Mobile IP wie beispielsweise Routenoptimierung oder Trigger durch niedere Schichten bleiben unberücksichtigt, da diese in heutigen Systemen typischerweise nicht eingesetzt werden können.

Die Anwendungsschicht ist hier repräsentiert durch den WWW-Browser im mobilen Terminal einerseits und einen auf dem Foreign Agent angeordneten HTTP-Proxy (*Squid* [Cha05]) anderer-

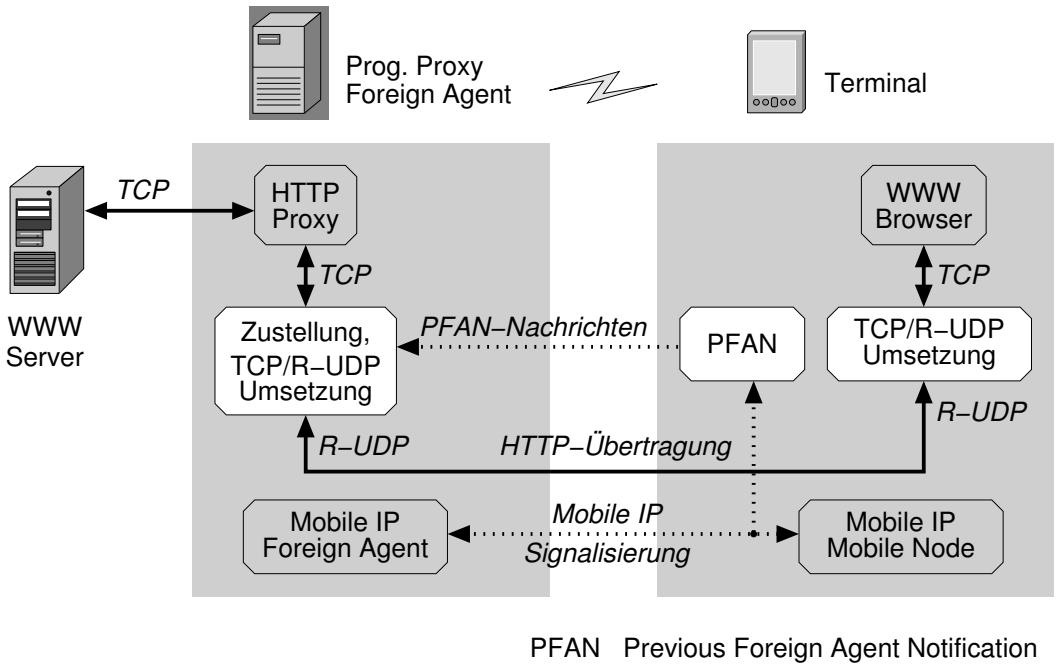


Abbildung 6.5: Softwarestruktur der prototypischen Implementierung.

seits. Die TCP-Verbindung zwischen Browser und Proxy wird in eine gesicherte UDP-basierte Übertragung – genannt Reliable UDP (R-UDP) – umgesetzt, wodurch sich besonders einfach eine eigene, angepaßte Flußkontrolle realisieren läßt.

Die *Previous Foreign Agent Notification* (PFAN) benachrichtigt den vorherigen Foreign Agent. Die gleichnamige Einheit erkennt anhand der Mobile IP-Registrierungsnachrichten einen vollzogenen Wechsel des Foreign Agent und sendet umgehend eine PFAN-Nachricht mit der IP-Adresse des vorherigen Foreign Agent an den aktuellen Foreign Agent. Diese wird dort von der Zustelleinheit – die zugleich die TCP/R-UDP-Umsetzung vornimmt – entgegengenommen. Die Zustelleinheit wird dann die Partnerinstanz auf dem vorherigen Foreign Agent benachrichtigen und die Weiterleitung der Daten veranlassen. Aufgrund der Vereinigung der vereinfachten Funktionen der Mobilitätssteuerung und des höheren Mobilitätsrouting einerseits, sowie der Flußkontrolle für die Datenströme zum mobilen Terminal andererseits, läßt sich ein Handovertrigger für die Fußkontrolle besonders einfach realisieren und so die Wiederaufnahme der Datenübertragung zum mobilen Teilnehmer beschleunigen.

6.3.2 Benachrichtigung des vorherigen Foreign Agent (PFAN)

Bei Mobile IP gemäß [Per02] wird der vorhergehende Foreign Agent nicht darüber informiert, wann und wohin ein mobiler Teilnehmer gewechselt ist. Auch wird dem neuen Foreign Agent nicht mitgeteilt, welcher der zuletzt besuchte Foreign Agent war. Es gibt jedoch eine Reihe von Erweiterungen zu Mobile IP, die diese Information bereitstellen. Die in Abschnitt 2.2.5

vorgestellte *Post-Registrierung* [EM05] baut zwischen dem vorherigen und dem aktuellen Foreign Agent einen bi-direktionalen Tunnel auf, um Paketverluste beim Handover zu verringern. Allerdings verläßt sich diese Methode darauf, die dazu benötigte Adresse des vorherigen Foreign Agent aus einem Trigger der Sicherungsschicht zu erhalten. Im Rahmen des ebenfalls bei der IETF vorgeschlagenen Kontexttransfers [LNPK05] wird die Adresse des vorherigen Foreign Agent vom mobilen Terminal im *Context Transfer Activate Request* (CTAR) an den neuen Foreign Agent übermittelt, siehe auch 5.1.1. Im Falle der in Abschnitt 2.2.4 vorgestellten Routenoptimierung muß ebenfalls der vorherige Foreign Agent benachrichtigt werden, wobei in [PJ98] vorgeschlagen wird, dazu die *Binding Update*-Nachricht zu verwenden, wie sie nach einem Handover auch den Kommunikationspartnern zugestellt wird.

Für die prototypische Implementierung sollen das Mobile IP-Protokoll und die Implementierungen seiner einzelnen Teile unverändert bleiben. Daher wird eine eigene *Previous Foreign Agent Notification* (PFAN)-Nachricht anstatt einer Erweiterung der reguläre Mobile IP-Registrierungsnachricht vom mobilen Terminal an den Programmierbaren Proxy verschickt. Da die verwendete Mobile IP-Implementierung keine entsprechenden Schnittstellen bietet, greift die PFAN-Instanz im mobilen Terminal die Registrierungspakete von Mobile IP über einen *Packet Socket* ab, wie in [Sch02] beschrieben. Bei jeder positiv beschiedenen Registrierungsantwort eines Foreign Agent prüft die PFAN-Instanz des mobilen Terminals, ob diese vom bisherigen oder einem neuen Foreign Agent stammt. Ist letzteres der Fall, stellt die Registrierungsnachricht nicht lediglich eine Erneuerung der aktuellen Registrierung dar, sondern kann als Beleg für einen erfolgreichen Netzwechsel gewertet werden. Wie in Abb. 6.6 dargestellt, wird dies als Trigger für das Versenden der PFAN-Nachricht an den neuen Foreign Agent verwendet, der daraufhin umgehend die Datenzustellung an den mobilen Teilnehmer einleitet.

Wie bei den oben genannten Verfahren die einer ähnlichen Mitteilung bedürfen, muß auch diese Nachricht authentifiziert werden, um mobile Teilnehmer davor zu bewahren, daß böswillige Angreifer einen Kontexttransfer veranlassen und so einen mobilen Teilnehmer seiner Datenströme und netzseitigen Dienste “berauben”. Da dies mit Standardverfahren lösbar ist, soll in dieser Arbeit nicht weiter darauf eingegangen werden.

Bei diesem Prototyp gibt es nur einen einzigen mobilen Teilnehmer mit einem einzigen genau bestimmten Dienst (den HTTP-Proxy). Daher ist es für die Funktionsweise unerheblich, ob der Teilnehmer mit der PFAN-Nachricht zugleich mitteilt, welche genau bestimmten Dienste am neuen Ort bereitgestellt werden sollen (wie dies beispielsweise beim Kontexttransferprotokoll CXTP der Fall ist, vgl. oben und Abschnitt 5.1.1) oder ob sich dies per se auf alle Dienste des mobilen Terminals bezieht.

6.3.3 R-UDP und Handover-getriggerte Flußkontrolle

Mit dem Testaufbau soll nicht nur die Architektur an sich, sondern auch der Vorteil der Handover-getriggerten Flußkontrolle verifiziert werden. Da jedoch das TCP fest im Betriebssystem verankert ist, wäre es sehr aufwendig, dieses für den Handovertrigger zu modifizieren. Stattdessen erfolgt die prototypische Implementierung der Datenübertragung zwischen HTTP-Proxy und

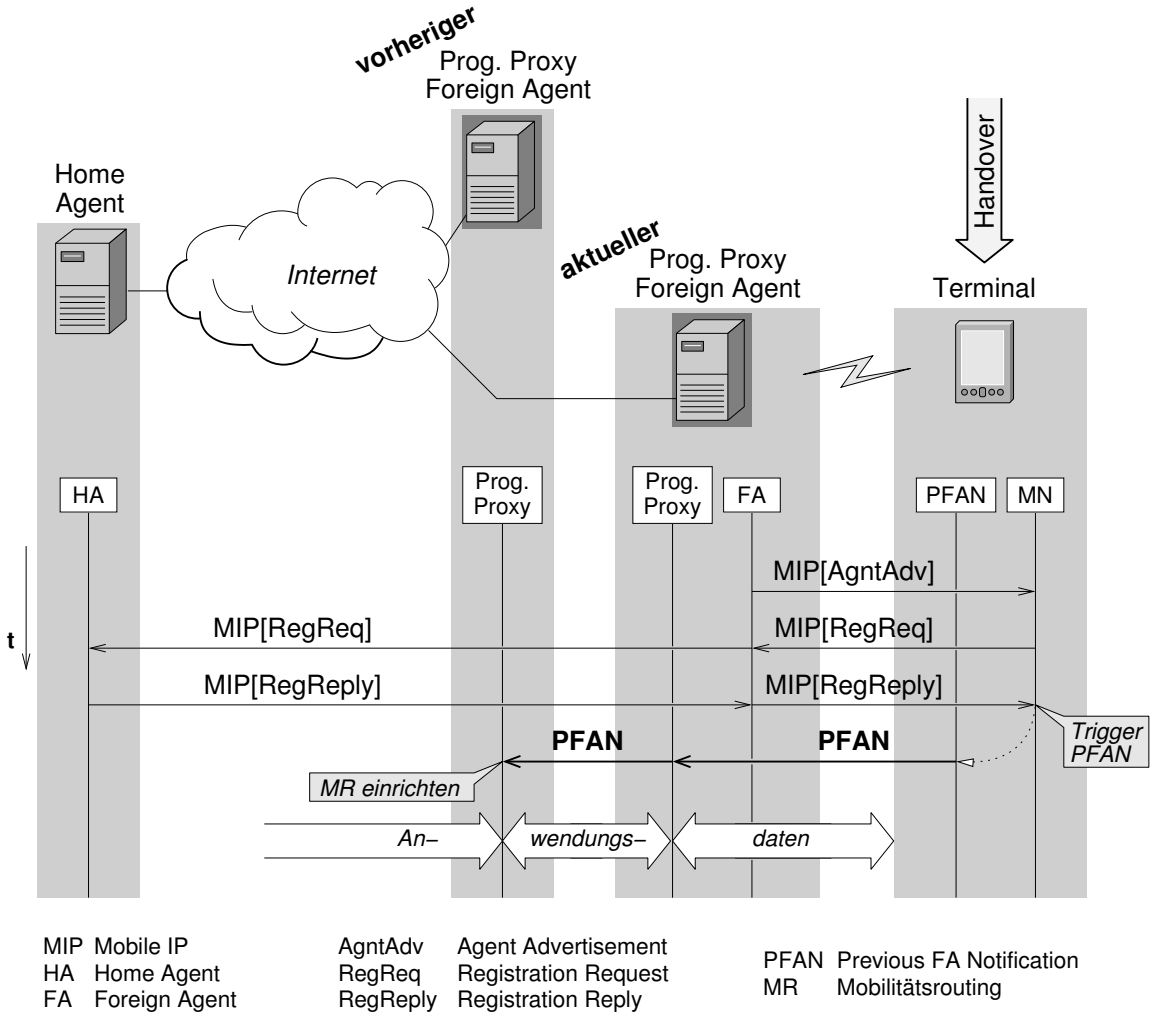


Abbildung 6.6: Nachrichtenflußdiagramm der Benachrichtigung des vorherigen Foreign Agent.

mobilem Terminal auf Basis von UDP, wodurch es vergleichsweise leicht ist, eine eigene Flußkontrolle mitsamt der Triggerung nach einem Handover zu implementieren.

Dazu wird auf UDP ein einfaches Protokoll – genannt *Reliable UDP* (R-UDP) – aufgesetzt, welches die Erkennung von Paketverlusten ermöglicht. Dieses fügt (analog zu TCP) in jedes UDP-Paket zwischen dem Header und den eigentlichen Daten sowohl eine Pakettypkennung (Datenpaket, Empfangsbestätigung, Verbindungsauf- und -abbau) als auch eine fortlaufende Sequenznummer zur Überwachung der Datenübertragung ein. Auf diese Weise bleibt die Ende-zu-Ende-Semantik der HTTP-Transaktion erhalten.

In Abb. 6.7 sind der Verbindungsaufbau, die HTTP-Anfrage und die anschließende HTTP-Datenübertragung zwischen den beteiligten Instanzen[4] dargestellt. Der Verbindungsabbau geschieht

[4]Der Home Agent wurde der Übersichtlichkeit halber weggelassen, da er hier bei der Datenübertragung nicht beteiligt ist.

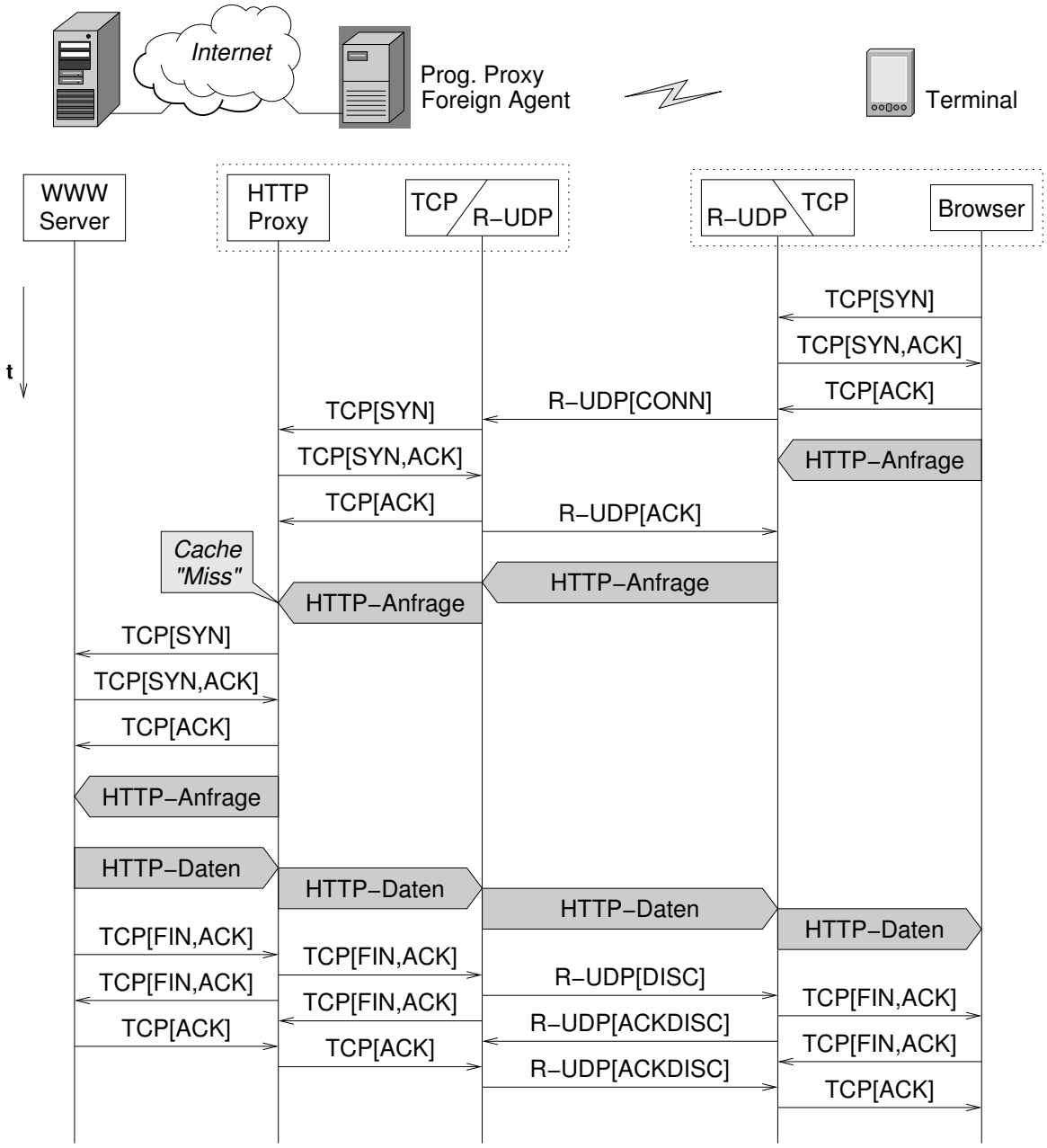


Abbildung 6.7: Verbindungsaufbau und HTTP-Datenübertragung mit R-UDP.

analog dem Verbindungsaufbau in umgekehrter Reihenfolge. Die Komplexität vom TCP wird dabei jedoch vermieden, da R-UDP ausschließlich das betrachtete Downloadszenario nachbildet, was einige vereinfachende Annahmen für die prototypische Implementierung gestattet. Aufgrund der kurzen und konstanten Paketumlaufzeit zwischen den beiden Endpunkten kann auf Mechanismen zu deren Schätzung verzichtet werden. Auch wird die Datenrate des Protokolls fest eingestellt, indem eine konstante Fenstergröße vorgegeben wird. Auch wird angenommen, daß die Reihenfolge der Datagramme beibehalten wird und Paketverluste hier nur durch die Handovervorgänge ausgelöst werden. Dies gestattet einen einfachen *go-back-n*-Ansatz, bei dem doppelte Bestätigungen bereits als Indiz für Paketverluste gewertet werden und eine Wiederholung des gesamten Sendefensters auslösen. Dies entspricht insofern dem eingeschwungenden Zustand vom TCP, als dieses sich nach dem Starten auf eine annähernd konstante Datenrate einregelt, die durch den *Bottleneck Link* der Verbindung bestimmt ist.

Nach einem Handover und der folgenden Benachrichtigung des Programmierbaren Proxy wird die Übertragung unverzüglich mit dieser ursprünglichen Datenrate wiederaufgenommen. Sind die Verhältnisse im neuen Netz vergleichbar, wird damit der eingeschwungene Zustand des TCP vorweggenommen. Da jedoch kein *Slow Start* vorgenommen wird sondern die Sendefenstergröße unverändert bleibt, wird nach dem Handovertrigger – wie beim *hot-TCP* – sofort ein vollständiges Sendefensters losgeschickt. Wie in Abschnitt 6.4.4 beschrieben, beeinträchtigt dieses Verhalten jedoch nicht die Fairneß zwischen verschiedenen TCP-Verbindungen.

6.4 Evaluierung des Programmierbaren Proxy

6.4.1 Konzeption des Meßaufbaus und Meßmethodik

Für die Evaluierung wird der in Abb. 6.8 dargestellte Testaufbau verwendet. Er besteht aus mehreren IP-Netzen, die über Router miteinander verbunden sind. Das mobile Terminal ist über eine von ihm gesteuerte *Bridge*[5] mit den Netzen der beiden Foreign Agents verbunden. Dadurch kann das Terminal den Zeitpunkt des IP-Netzwechsels exakt vorgeben und für die Messung auch korrekt erfassen. Die Signallaufzeiten der Leitungen sind gleich und können für den Versuch vernachlässigt werden. Als Datenquelle dient ein über ein hochbitratiges Datennetz angeschlossener WWW-Server.

Auf den einzelnen Rechnern werden – neben den entsprechenden Mobile IP-Instanzen – die bereits in Abb. 6.5 vorgestellten Einheiten gestartet. Auf den Foreign Agents sind dies zusätzlich der HTTP-Proxy und die Zustelleinheit mit der TCP/R-UDP-Umsetzung, auf dem mobilen Terminal die TCP/R-UDP-Umsetzung, die PFAN-Einheit sowie ein Hilfsprogramm zur Steuerung der *Bridge* und zur Aufnahme der Meßwerte. Der WWW-Server ist ein regulärer *Apache* HTTP-Server [Apa]. Die Mobile IP-Instanzen versenden ihre *Agent Advertisement*-Nachrichten im Abstand von nominell 1 s, dem minimalen vorgesehenen Wert. Tatsächlich wird dieser Ab-

[5] Sowohl der Router als auch die *Bridge* sind PC-basiert.

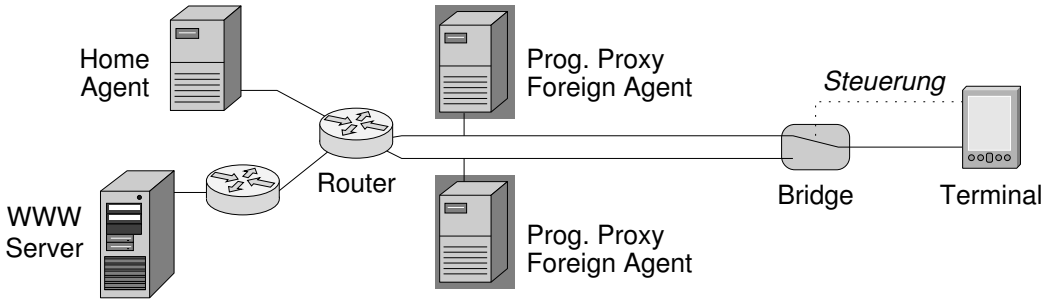


Abbildung 6.8: Versuchsaufbau für die Messung zur Evaluierung des Programmierbaren Proxy.

stand jedoch gemäß [Per02] von den Systemen leicht variiert, um Synchronisierungseffekte zu vermeiden.

Um derartige Synchronisierungen von Seiten des Clients zu vermeiden, wird der Abstand zwischen zwei Handovern stets als nicht ganzzahliges Vielfaches dieses nominellen Abstandes zweier *Agent Advertisements* gewählt. Dadurch wird eine Gleichverteilung der Zeiten zwischen dem Wechsel in ein Netz und dem Eintreffen des ersten *Agent Advertisements* über die verschiedenen Messungen erreicht.

In Abb. 6.9 ist der HTTP-Durchsatz des Programmierbaren Proxy über der Handoverrate aufgetragen. Zum Vergleich wird der Durchsatz von regulärem TCP über den normalen Mobile IP-Mechanismus gemessen. Bei den Meßpunkten sind jeweils die 99,9%-Konfidenzintervalle angetragen. Jedem Meßpunkt liegen mindestens zehn Meßwerte zugrunde (bei hoher Varianz dieser Meßwerte bis zu 100), um für alle Meßpunkte einer Kurve vergleichbare Konfidenzmaße zu erhalten. Gemessen wird stets der HTTP-Durchsatz je Handoverintervall am Empfänger nach Abzug aller unterhalb des HTTP liegenden Protokollinformationen und Sendewiederholungen. Dementsprechend wird dabei nur die jeweils erste korrekte Übertragung eines Datenpaketes berücksichtigt.

6.4.2 Analyse des Verhaltens von TCP mit Mobile IP

Ohne Mobilität liegt der Durchsatz des regulären[6] TCP geringfügig über dem des Programmierbaren Proxy. Dies läßt sich daraus erklären, daß beim TCP die Daten Ende-zu-Ende übertragen werden, während im Falle des Programmierbaren Proxy zusätzlich noch die Umsetzung in R-UDP (und umgekehrt) erfolgt und der HTTP-Proxy zwischengeschaltet ist. Der Durchsatz von TCP fällt dann mit steigender Handoverrate linear ab. Der leichte Knick vor dem untersten Meßpunkt ist darin begründet, daß der Durchsatz bereits bei etwa 9 Handovervorgängen je Minute – also vor dem letzten Meßpunkt – praktisch auf Null abfällt. Dies geht auch aus den in Abb. 6.10 dargestellten Handoverunterbrechungsdauern hervor. Die Unterbrechungsdauer ist dabei der Zeitabstand zwischen dem letzten Datenpaket vor dem Handover und dem ersten

[6] Aufgrund der beim Handover auftretenden Paketverlustcharakteristik bieten Verbesserungen wie *Fast Retransmit* oder *Selective Acknowledge* (SACK) hier keinerlei Vorteile.

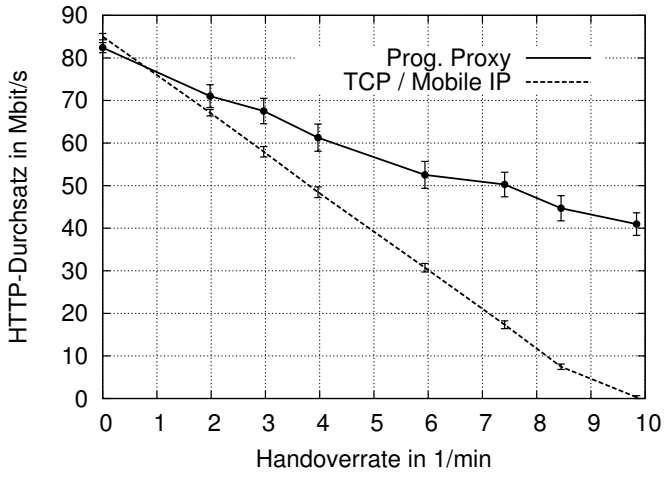


Abbildung 6.9: Vergleich des HTTP-Durchsatz von Programmierbarem Proxy und TCP mit Mobile IP in Abhängigkeit von der Mobilität des Teilnehmers (mit 99,9% Konfidenzintervallen).

Datenpaket nach einem Handover, wobei auch hier Wiederholungen bereits korrekt empfangener Pakete nicht berücksichtigt werden. Deutlich ist zu erkennen, daß das TCP nach einem *Retransmission Timeout* erst nach festen, zunehmend länger werdenden Intervallen einen erneuten Sendeversuch unternimmt.

Unterschreitet der Handoverabstand den Wert des Großteils der Unterbrechungsdauern (6,4 s), werden nur in einem entsprechend geringen Anteil der Handoverintervalle überhaupt noch einige wenige Datenpakete übertragen. Bedingt durch den *Slow Start*-Mechanismus von TCP geschieht dies außerdem nur mit verringerter Sendefenstergröße. Daher fällt die Datenrate bei Verwendung von TCP zunächst mit einer solchen Steigung ab, daß sie die Abszisse beim Mittelwert aller

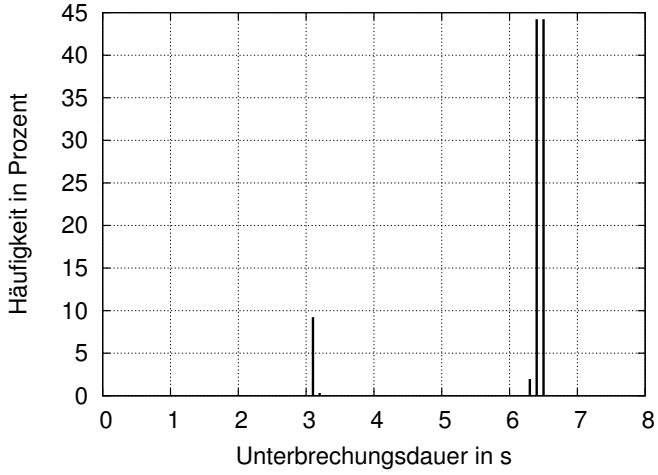


Abbildung 6.10: Verteilung der Unterbrechungsdauern bei TCP im Fall eines Handover (ca. 300 Meßwerte, gerundet auf Zehntelsekunden).

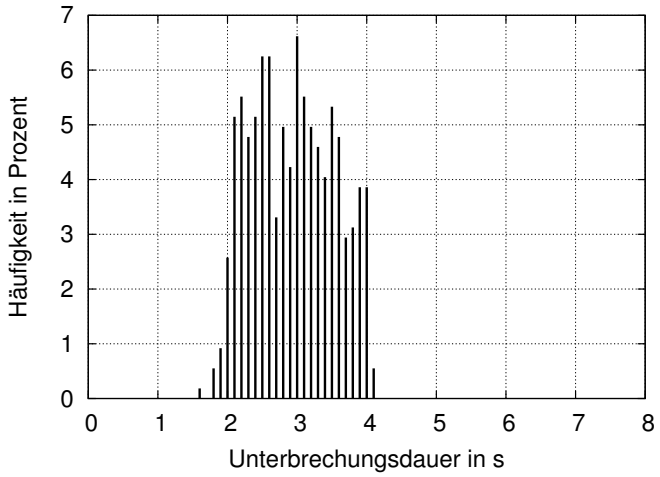


Abbildung 6.11: Verteilung der Unterbrechungsdauern bei R-UDP im Fall eines Handover (ca. 500 Meßwerte, gerundet auf Zehntelsekunden).

Handoverunterbrechungsdauern schneiden würde. Kurz davor verflacht die Kurve jedoch und erst wenn das Handoverintervall kürzer als die minimale Unterbrechungsdauer ist, ist tatsächlich keinerlei erfolgreiche Datenübertragung mehr möglich.

6.4.3 Analyse des Verhaltens des Programmierbaren Proxy mit R-UDP

Durch den im Rahmen der vorgeschlagenen Architektur des Programmierbaren Proxy ermöglichten Handovertrigger nach Vollendung des Handover verringert sich der Nettodurchsatz des R-UDP im Vergleich zu dem des TCP bei weitem nicht so stark. Da die Flußkontrolle nach dem Handover sofort mit dem Senden beginnt, kann sie die zur Verfügung stehende Zeit zwischen der Wiederherstellung des Routings bis zum nächsten Handover fast vollständig nutzen. Bei etwa zehn Handovervorgängen pro Minute, was einem Handoverintervall von 6 s entspricht, erreicht das R-UDP immer noch die Hälfte der Datenrate im statischen Fall, während hingegen das TCP bei dieser Handoverrate bereits nur noch sporadisch Daten überträgt.

Approximiert man die Meßwerte mittels einer Geraden, liegt deren Schnittpunkt mit der Abszisse im Bereich um 20 Handover je Minute. Dies entspricht dem Mittelwert der im Histogramm in Abb. 6.11 aufgetragenen Handoverunterbrechungszeiten.

Da R-UDP nach dem Handovertrigger unverzüglich mit dem Senden beginnt, entspricht die Unterbrechungsdauer genau der Zeit, die Mobile IP für die Wiederherstellung des IP-Routings benötigt. Der theoretische Minimalwert dafür wäre 2 s, da Mobile IP drei *Agent Advertisements* des neuen Foreign Agent abwartet, bevor die Registrierung bei diesem eingeleitet wird. Daß in seltenen Fällen dennoch geringfügig kürzere Unterbrechungszeiten bis zu 1,6 s auftreten liegt daran, daß der Abstand zwischen den *Advertisements* zufällig variiert wird. Andererseits treten aus diesem Grunde auch Zeiten bis zu 4,1 s auf.

Mit zunehmender Entfernung des Home Agent verlängert sich der Routingpfad für die Datenpakete im herkömmlichen Fall. Der Vorteil des Programmierbaren Proxy gegenüber TCP über Mobile IP erhöht sich jedoch, je weiter der Home Agent entfernt ist. Dies äußert sich sowohl in einer verringerten oberen Schranke für die TCP-Datenrate (diese ist dadurch gegeben, daß je Paketumlaufzeit höchstens einmal die Sendefenstergröße übertragen werden kann) und natürlich auch kürzeren Antwortzeiten, vergleiche [**TBL03**].

6.4.4 Simulative Untersuchung der Fairneß von hot-TCP

Beim TCP wird die Sendedatenrate nur mittelbar über die Sendefenstergröße reguliert. Setzt man nun beim Handovertrigger die Größe des Sendefenster auf den Wert vor dem Handover zurück, sendet der Proxy sofort sein ganzes Sendefenster unbestätigter TCP-Segmente an den entsprechenden Teilnehmer aus. Da die Funkstrecke eine deutlich geringere Datenrate aufweist als das LAN zwischen dem Proxy und der Basisstation, treffen diese als *Burst* in der Warteschlange der Basisstation ein. Typischerweise sind die Puffer der Basisstationen so dimensioniert, daß sie in der Lage sind ein maximales TCP-Fenster aufzunehmen. Es kann angenommen werden, daß der *Burst* im Pufferspeicher Platz findet, wenn zur Verwaltung der Warteschlange die *Random Early Detection* (RED) [FV93] verwendet wird.

Der RED-Algorithmus hält den durchschnittlichen Füllstand eines Pufferspeichers niedrig, indem bereits bei einem vergleichsweise geringen Ansteigen des Pufferfüllstandes zufällig einige Datagramme verworfen werden. Dadurch wird die TCP-Flußkontrolle früh gedrosselt. Füllt nun ein nach dem Handovertrigger ausgesandter *Burst* den Pufferspeicher auf, werden Pakete des *Burst* mit höherer Wahrscheinlichkeit verworfen als Pakete anderer Verbindungen. Daher wird auch die Verbindung, die nach einem Handovertrigger mit einer zu hohen Datenrate sendet, mit hoher Wahrscheinlichkeit diejenige sein, die ihre Datenrate am deutlichsten verringert. Da diese Verringerung proportional zur Größe des Sendefensters ist, trägt diese Verbindung dann auch signifikant zur Verringerung der eventuellen Überlastsituation bei. Daraus resultiert, daß andere Teilnehmer durch *hot-TCP* nicht benachteiligt werden.

Diese Folgerung wird durch die in [**TSSB04**] dargelegten simulativen Untersuchungen belegt. Die Simulationen, für die der *Network Simulator* (ns-2) verwendet wird, nehmen die in Abb. 6.12 gezeigte Topologie an. Die Funkübertragungsstrecke, die zugleich den Engpaß bei der Datenübertragung zwischen Proxy und Terminals darstellt, wird vereinfachend als eine feste Verbindung mit beschränkter Datenrate modelliert. Dies ist legitim, da Paketverluste bei der drahtlosen Übertragung durch Sendewiederholungen auf der Sicherungsschicht ausgeglichen werden und sich die Funkübertragungsstrecke daher auch wie eine Festnetzleitung verhält, deren Datenrate gleich der Kanalkapazität ist [BPSK97].

In dem untersuchten einfachen Downloadszenario befinden sich zwei Teilnehmer in einer Funkzelle und empfangen mittels TCP Daten vom Proxy mit der maximal verfügbaren Datenrate. Das dritte Terminal nutzt ebenfalls die maximal verfügbare Datenrate für seine Übertragung. Abb. 6.13 zeigt die Aufteilung der Übertragungskapazität des Funkkanals nach dem Handover des dritten Terminals in die von den beiden anderen Terminals benutzte Funkzelle.

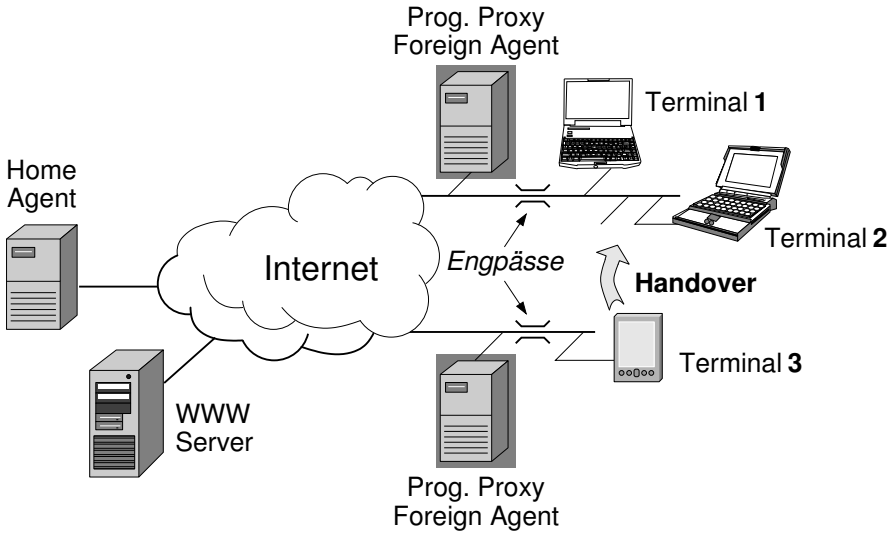


Abbildung 6.12: Topologie für die Simulation des Verhaltens von *hot-TCP*.

Beim regulärem TCP[7] erhöht sich der *Retransmission Timer* durch die Unterbrechung der Datenübertragung beim Handover gemäß dem *Backoff*-Algorithmus. Daher setzt reguläres TCP die Datenübertragung erst nach einer unnötig langen Wartezeit fort, die durch die mehrfache Vervielfachung der Länge des *Retransmission Timer* nach wiederholten Zeitüberschreitungen bedingt ist. Deutlich erkennbar ist der exponentielle Anstieg der Datenrate des Terminals 3 gemäß dem *Slow Start*. Diese Verzögerung stellt einen deutlichen Nachteil des regulären TCP dar, insbesondere im Falle von Echtzeitkommunikation oder wenn sich das mobile Terminal schnell von einer Basisstation zur nächsten bewegt.

Im Gegensatz dazu beginnt *hot-TCP* die Wiederholung der ausstehenden Datagramme sofort mit Erhalt des Handovertriggers nach dem Ende der Mobile IP-Registrierungsprozedur. Deutlich ist in Abb. 6.13 zu erkennen, daß Terminal 3 zunächst einen größeren Anteil an der Datenübertragungsrate beansprucht, da ein vollständiges Sendefenster auf einmal übertragen wird. Durch die oben beschriebenen Mechanismen drosselt *hot-TCP* jedoch umgehend seine Datenrate und erreicht nach einer kurzen Spitze sogar früher eine faire Verteilung als das reguläre TCP. Da das Terminal jedoch zuvor für einige Zeit (vom Zeitpunkt des Netzwechsels bis zum Ende der Mobile IP-Registrierungsprozedur) keinerlei Übertragungskapazität beansprucht hat, verschafft diese Bevorzugung nach einem Handover dem mobilen Terminal insgesamt gesehen keinen systematischen Vorteil gegenüber den statischen Terminals, so daß – global gesehen – die Fairneß von TCP durch *hot-TCP* nicht beeinträchtigt wird.

[7] Hier wurde die Variante *NewReno* [FHG04] verwendet. Diese unterscheidet sich in ihrem Handoververhalten nicht von den anderen gängigen TCP-Varianten, wie beispielsweise TCP SACK [FMMP00], wenn der Handover zu einem *Retransmission Timeout* führt.

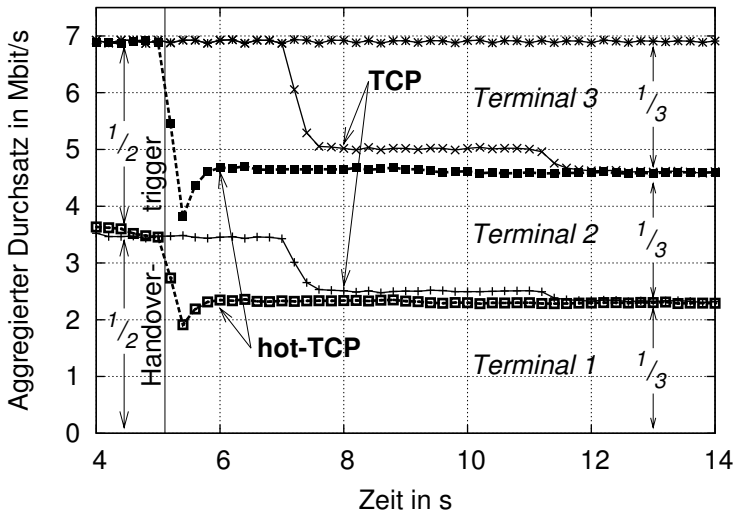


Abbildung 6.13: Vergleich der Aufteilung der Datenrate des Funkkanals bei regulärem TCP und *hot-TCP* nach dem Handover eines Teilnehmers.

6.5 Fazit

Mit dem *Programmierbaren Proxy* wurde ein mobiler programmierbarer Dienst entworfen und prototypisch implementiert. Dies erfolgte auf der Grundlage der im vorangehenden Kapitel dargelegten Architektur, wobei die Realisierung auf die zur Evaluierung dieses Dienstes erforderlichen Elemente reduziert wurde, also den *kritischen Pfad* des Protokolldatenflusses. Die Flußkontrolle ist bei der Implementierung des *hot-TCP* insofern vereinfacht, als vom eingeschwungenen Zustand des TCP ausgegangen wird.

Der Programmierbare Proxy verknüpft die Flußkontrolle mit der verfügbaren Mobilitätsinformation. Nach einem Handover wird so augenblicklich die Sendewiederholung der noch unbestätigten Datagramme angestoßen. Außerdem wird die Datenrate auf den Wert vor dem Handover eingestellt, so daß der *Slow Start* entfällt. Verglichen mit TCP – ebenfalls mit darunterliegendem Mobile IP – wird durch diese beiden Maßnahmen der Datendurchsatz deutlich erhöht, insbesondere im Falle häufiger Handover.

Die vorgenommenen Messungen zeigen die Effizienz dieser Maßnahmen. Der Datendurchsatz nimmt – nach Abzug von Sendewiederholungen – unter Verwendung von regulärem TCP doppelt so schnell mit der Handoverrate ab. Ab einer gewissen Handoverrate ist mit regulärem TCP keinerlei Datenübertragung mehr möglich, während hingegen der Programmierbare Proxy noch die Hälfte des maximalen Durchsatzes gestattet.

Kapitel 7

Zusammenfassung und Ausblick

In der vorliegenden Arbeit werden die Aspekte von Mobilität und programmierbaren Diensten übergreifend und umfassend behandelt. Programmierbare Dienste werden dynamisch auf dedizierten Knoten innerhalb des Zugangsnetzes geladen und mit der Mobilitätsunterstützung des Teilnehmers gekoppelt. Betrachtet werden beide Aspekte der Mobilität, *Roaming* und *Handover*.

Vorrangiges Ziel bei dieser Arbeit war der Entwurf einer praktikablen, also mit vertretbarem Aufwand in heutigen Netzen realisierbaren Lösung. Vergleichsweise einfach zu bewerkstelligen sind Änderungen an den mobilen Terminals selbst sowie im Bereich der *Zugangsnetze* beziehungsweise -domänen, während hingegen Modifikationen an den entfernten Teilnehmern und in den *Transportnetzen* weitgehend ausgeschlossen sind. Dieses Ziel wurde mit der vorgeschlagenen Architektur erreicht.

Ausgangspunkt für die erarbeitete Architektur sind die existierenden Verfahren zur Mobilitätsunterstützung in IP-basierten Netzen. Neben Mobile IP – der wohl prominentesten Lösung – und den wichtigsten Verbesserungen beziehungsweise Erweiterungen werden weitere Ansätze zur Mobilitätsunterstützung vorgestellt und miteinander verglichen. Wesentliche Kriterien für den Vergleich der Verfahren zur Makro-Mobilitätsunterstützung sind dabei die Art der Anforderung an die netzseitige Unterstützung und das Verhalten beim Handover. Zu berücksichtigen ist dabei sowohl die Reaktion beim unantizipierten *break-before-make* Handover als auch dem vorab signalisierten *make-before-break* Handover, wie er beispielsweise für vertikale Handover in Overlaynetzen typisch ist.

Um ein erschöpfendes Bild der Mobilitätsaspekte mobiler Teilnehmer zu bieten, erfolgt in dieser Arbeit zunächst ein Blick “nach unten” auf die Eigenschaften von Wireless LAN in Kapitel 3. Einen Blick “nach oben” auf die programmierbaren Netzdienste sowie die zugehörigen Plattformen gewährt Kapitel 4. Schließlich wird in Kapitel 5 eine Architektur für transparente mobile programmierbare Dienste vorgestellt, die beim Handover den “Blick zurück” zum vorherigen Netzzugang und auf diese Weise eine Steigerung der Handoverperformanz ermöglicht. Zuletzt bietet Kapitel 6 einen Blick “hinein” in die Realisierung eines mobilen programmierbaren Dienstes, komplementiert durch einen mit simulativen Methoden gewonnenen Blick “von

oben" auf die resultierenden Systemeigenschaften. So erschließt diese Arbeit ein vollständiges Panorama der Mobilitätsunterstützung mit Programmierbaren Netzen.

7.1 Überblick über die Ergebnisse

Die Analyse des systemrelevanten Verhaltens des Wireless LAN nach dem Standard IEEE 802.11 beginnt mit der Erkenntnis, daß die Kanalauswahlstrategie eine *Hysterese* erzeugt, die in der bislang üblichen analytischen Modellierung nicht enthalten ist. Bei genauerer Betrachtung stellt sich heraus, daß diese Hysterese sowohl durch die Schwellwerte der Empfangspegel als auch die geometrischen Gegebenheiten der Netztopologie bedingt ist, wobei letzteres der bestimmende Faktor ist. Die weitergehenden geometrischen Betrachtungen offenbaren, daß sich eine Mindestdistanz zwischen zwei Handovern bestimmen läßt. Diese läßt sich direkt in eine von der Geschwindigkeit des Teilnehmers abhängige *Mindestdauer* zwischen zwei Handovern umrechnen. Die praktische Relevanz dieser Verbesserung gegenüber der bislang verwendeten Modellierung ergibt sich bei einem Abgleich mit der Dauer der Mobile IP-Handoverprozedur: Bei den gleichen idealisierenden Annahmen (reguläre Anordnung der Basisstationen und homogene Funkwellenausbreitungsbedingungen) ergibt sich mit der hier vorgeschlagenen detaillierteren Modellierung eine deutlich geringere Wahrscheinlichkeit für abgebrochene Mobile IP-Handover.

Die auf Basis einer idealisierten programmierbaren Plattform entworfene *flexible Architektur* für mobile programmierbare Netzdienste ermöglicht es, mobilen Teilnehmern modular aufgebaute Netzdienste auf effiziente Art und Weise bereitzustellen. Die Architektur zeichnet sich dadurch aus, daß einerseits die mobilitätsbezogenen Funktionen vom zentralen Dienstmanagement an den jeweiligen Aufenthaltsort des Teilnehmers ausgelagert werden. Dies gestattet zusammen mit der direkten Kopplung an die Mobilitätsunterstützung des Terminals – beispielsweise Mobile IP – eine unmittelbare Reaktion auf einen Handover des mobilen Teilnehmers. Andererseits bietet die Architektur auch Flexibilität hinsichtlich dieser Reaktion, indem ein *Diensttransfer* des betreffenden Dienstes durchgeführt wird oder aber der Dienst an seinem ursprünglichen Ort verbleibt und die entsprechenden Anwendungsdatenströme mittels *Datentunnelung* dorthin zurückgeleitet werden. Der Diensttransfer nutzt dabei der Methode des Kontexttransfers. Dessen bei der IETF spezifiziertes Konzept wurde dazu dahingehend erweitert, daß auch transparente Dienste unterstützt werden, die beispielsweise von einem Administrator für das mobile Terminal eingerichtet werden.

Ob nach einem Handover des Teilnehmers ein Dienst transferiert oder lediglich die Datentunnelung dafür eingerichtet wird, hängt vom Erwartungswert der Summe des Aufwandes bis zum darauffolgenden Handover ab. Zur Bewertung dieses Aufwandes wird eine normierte Kostenmetrik eingeführt und die *Break-Even-Zeit* definiert als derjenige Zeitpunkt nach einem Handover, zu dem die kumulierten Kosten für Diensttransfer und Datentunnelung gleich hoch sind. Die Break-Even-Zeit bestimmt sich aus der Größe des Dienstkontextes und der Anwendungsdatenrate, so daß diese Zeit eine inhärente Eigenschaft einer jeden Anwendung und des damit verbundenen Dienstes ist. Vollzieht sich der nächste Handover vor Ablauf der Break-Even-Zeit, so ist

die Datentunnelung effizienter, andernfalls der Diensttransfer. Die hier vorgeschlagene Einheitsstrategie verwendet für diese Entscheidung den Erwartungswert der Zeit zum nächsten Handover bezüglich aller Teilnehmer, ohne weiter zwischen diesen zu differenzieren. Als Basis dafür dient die in Kapitel 3 ermittelte Verteilung der Zwischenhandoverzeiten. Ein Vergleich mit dem theoretischen Kostenoptimum ergibt, daß die Einheitsstrategie derart nahe daran heranreicht, daß der Aufwand für eine weitere Verbesserung in diesem Punkt – mittels individueller Bewegungsprädiktion – kaum lohnend scheint.

Die prototypische Implementierung eines programmierbaren Dienstes beinhaltet alle für den *kritischen Pfad* des Protokolldatenflusses wichtigen Elemente. Mit dem *Programmierbaren Proxy* wurde dabei ein Dienst realisiert, der die Datenzustellung für mobile Teilnehmer verbessert, indem die Datagramme nach einem Handover lokal an den mobilen Teilnehmer weitergeleitet werden. Dabei sorgt die im Proxy enthaltene Instanz des *Handover-Triggered TCP* (hot-TCP) dafür, daß die Flußkontrolle nach Fertigstellung des Handover unverzüglich die Wiederholung unbestätigter Datagramme aufnimmt. Der Prototyp zeigt nicht nur in der durchgeführten Vergleichsmessung einen deutlich verbesserten Datendurchsatz gegenüber herkömmlichem TCP, sondern gleichzeitig auch die Gangbarkeit des Konzeptes der programmierbaren Dienste.

7.2 Ausblick

Gegenwärtig läßt sich in der Telekommunikationswelt eine Tendenz zur Konvergenz beobachten, die sich sowohl im Bereich der Netze, der Endgeräte und auch der Dienste abspielt. Im Zuge der Konvergenz werden die Funktionen mehrerer Systeme auf ein einziges abgebildet. Dies geschieht sowohl mit dem Ziel, den Aufwand – und somit Kosten – zu reduzieren als auch um neuartige Dienste einzuführen.

Die *Konvergenz der Netze* läßt sich festmachen an der Entwicklung der verschiedenen Telekommunikationssysteme zu IP-basierten Netzen gemäß dem *All IP*-Ansatz [PD00, MT02]. Dabei kommen sowohl im Kernnetz als auch im Funknetzbereich IP-Technologien zum Einsatz. Gemäß der Definition des *3rd Generation Partnership Project* (3GPP) ist die Verwendung des IP zwar die Basis, aber doch nur einen Teilaspekt des *All IP* [3GP05]. Von gleichrangiger Bedeutung ist, daß die bislang benutzten verschiedenen Protokolle und Verfahren – beispielsweise für Verschlüsselung, Vergebührung und Ressourcenreservierung – vereinheitlicht werden. Dies verringert den Entwicklungsaufwand der Hersteller und senkt gleichzeitig die Betriebskosten der Betreiber, da nicht mehr verschiedenartige Systeme nebeneinander existieren. Auch verlieren die Mobilfunknetze ihren Sonderstatus, da die Unterstützung mobiler Teilnehmer im Zuge der *Fixed Mobile Convergence* zu einer integralen Funktion des Netzes wird. Schließlich vereinheitlicht sich auch die Grundlage für die Entwicklung und Bereitstellung von Telekommunikationsdiensten beziehungsweise Mehrwertdiensten, was den Weg für ein Konvergieren von Diensten bereitet.

Diese *Konvergenz der Dienste* gestattet die Integration von Funktionen, die vormals aufgrund der Verschiedenheit der Technologien spezialisierte und oftmals auch proprietäre Lösungen erforderten. Ein Beispiel dafür ist die *Computer Telephony Integration* (CTI) [Che95] zur Verknüpfung

von Arbeitsplatzrechner und Fernsprecher. Demgegenüber ist der *All IP*-Ansatz gekennzeichnet durch vereinheitlichte Plattformen wie dem *Softswitch* [Ebe06] sowie offene Schnittstellen und Protokolle, die zumeist von der IETF spezifiziert werden. Netzspezifische Aspekte treten bei der Entwicklung neuer Dienste zunehmend in den Hintergrund, was die Verwendung höherer Programmiersprachen gestattet. Daher konnten Funktionen Realität werden wie beispielsweise die Steuerung von Telekommunikationsdiensten durch den Teilnehmer über das WWW.

Im Zuge der *Konvergenz der Endgeräte* ist eine zunehmende Verschmelzung vormals getrennter Funktionen zu beobachten, wobei das resultierende Gerät neue innovative Funktionen gestattet. Beispielsweise ergibt das Zusammenfügen von Telephon und Photoapparat eine vernetzte Digitalkamera, mit der sich aufgenommene Bilder direkt verschicken lassen. Die spezifischen Eigenschaften solcher Endgeräte erfordern jedoch auch passende Unterstützung durch entsprechende Dienste von Netzseite. Die oben genannte Vereinheitlichung der Dienstentwicklung gestattet es, diese mit geringerem Aufwand an die jeweiligen Nutzungsumstände anzupassen, wie beispielsweise die Beschränkungen eines mobilen Endgerätes. Abhängig vom Einsatzbereich gibt es die unterschiedlichsten Varianten von Bedienanzeigen (hinsichtlich Größe, Auflösung, Farbumfang), Eingabegeräten (Touchscreen, Griffel, numerische oder alphanumerische Tastatur), Funktechnologien und Übertragungsprotokollen sowie der Rechenleistung und nicht zuletzt der Batteriekapazität. Die dazu erforderlichen Methoden reichen von der Skalierung von Bildern, beziehungsweise Videos, bis hin zu einer vollständigen (semantisch äquivalenten) Reorganisation der Datendarstellung, z. B. durch deren Aufsplittung und Einfügen einer angepaßten Menüführung [Chi06].

Konvergenz ist somit ein Schlüsselfaktor für die effiziente Bereitstellung neuer Telekommunikationsdienste. Es ist zu erwarten, daß mit der Vielzahl und dem Variantenreichtum zukünftiger Dienste auch der Bedarf an Lösungen wie den Aktiven und Programmierbaren Netzen zunimmt. Für die Forschung im Bereich der Telekommunikationsnetze wird – um einen Blick “nach vorn” zu wagen – der Aspekt der Selbstkonfiguration und -organisation weiterhin von hoher Bedeutung sein. Für die Betreiber der Telekommunikationsnetze besteht hier ein besonderes Interesse, da ein Senkung des Konfigurationsaufwandes sowohl den Aufbau und die Erweiterung der Netze als auch die Bereitstellung neuer Dienstmerkmale vereinfacht und so zu einer Kostenreduktion beiträgt.

Anhang A

Ergänzende statistische Betrachtungen

A.1 Wegstrecke eines neuen Terminals

Der Abstand Z_n, den ein neuer Teilnehmer von seinem Ursprungspunkt (also dem Ort an dem er sein Terminal eingeschaltet hat) zum Rand des Versorgungsgebietes der Basisstation zurücklegt, hängt sowohl von der Entfernung des Ursprungsortes von der Basisstation (bezeichnet mit ρ) als auch der relativen Richtung seiner Bewegung (Winkel θ, vgl. Abb. A.1) nach [HR86] wie folgt ab:

$$Z_n = \sqrt{R^2 - (\rho \sin\theta)^2} - \rho\cos\theta \tag{A.1}$$

Zunächst sei angenommen, daß die mobilen Teilnehmer über das gesamte Versorgungsgebiet einer Basisstation gleichmäßig verteilt sind (die entsprechenden Formelzeichen sind mit einem einfach Strich gekennzeichnet). Die Wahrscheinlichkeitsdichte der Entfernung ρ des Ursprungsortes von der Basisstation ist dann

$$f'_\rho(\rho) = 2\rho/R^2 \quad \text{für} \quad 0 \le \rho \le R \tag{A.2}$$

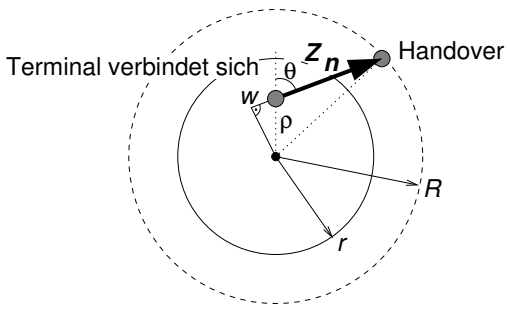


Abbildung A.1: Weglänge Z_n eines neuen Knoten zum ersten Handover.

Entsprechend dem angenommenen *Random Direction*-Mobilitätsmodell ist θ gleichverteilt in $[0..2\pi]$. Folglich läßt sich die Vorgehensweise in [HR86] ebenfalls für die Berechnung der Wahrscheinlichkeitsdichte der Wegstrecke Z'_n nutzen:

$$f'_{Zn}(z) = \frac{2}{\pi R^2}\sqrt{R^2 - \left(\frac{z}{2}\right)^2} \quad \text{für} \quad 0 \leq z \leq 2R \tag{A.3}$$

Tatsächlich ist jedoch bei dieser Verteilung der Entfernungen nicht berücksichtigt, daß Teilnehmer im Randbereich des Funkbereiches auch eine benachbarte Basisstation auswählen können. Da die Kanalauswahlstrategie von Wireless LAN nicht die einzelnen Feldstärken vergleicht, kann angenommen werden daß zufällig eine der empfangbaren Basisstationen ausgewählt wird. Folglich kann eine virtuelle Reduktion der Teilnehmerdichte im Kreisring zwischen r und R angenommen werden. Da jede empfangbare Basisstation mit gleicher Wahrscheinlichkeit ausgewählt wird, hängt die Wahrscheinlichkeit P_{AP} daß eine bestimmte Basisstation ausgewählt wird lediglich von der Anzahl der empfangbaren Basisstationen ab.

Die Wahrscheinlichkeit P_{AP} ist also $\frac{1}{2}$ in Bereichen mit doppelter und $\frac{1}{3}$ in Bereichen mit dreifacher Überlappung. Nimmt man P_{AP} als über die gesamte Fläche des Kreisringes konstant an, kann diese leicht über das Verhältnis der entsprechenden Flächen berechnet werden:

$$P_{AP} = \frac{1}{2} - \frac{1}{6(\frac{A_2}{A_3} - 1)} + \frac{1}{12}\frac{A_4}{A_2 - A_3} \tag{A.4}$$

Abhängig vom Verhältnis des Radius R und dem Zellabstand D liegt die Wahrscheinlichkeit P_{AP} zwischen 30,45% and 50%, wie in Abb. A.2 für D = 100 m gezeigt.

Die Symbole A_2 und A_3 bezeichnen wie in Gl. (3.6) beziehungsweise Gl. (3.7) die Flächen doppelter beziehungsweise dreifacher Überlappung von Funkzellen. Entsprechend steht A_4 für die Schnittfläche vierer benachbarter Zellen. Aufgrund der Regelmäßigkeit der hier angenommenen

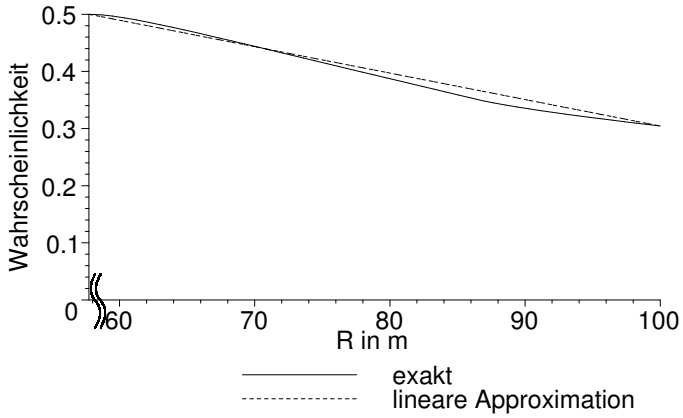


Abbildung A.2: Wahrscheinlichkeit P_{AP} im Kreisring zwischen R und r eine bestimmte Basisstation auszuwählen.

Topologie – die Fläche wird dabei nur durch zwei Kreislinien begrenzt – vereinfacht sich diese Fläche zu

$$A_4 = 2R^2 \arccos\left(\frac{\sqrt{3}D}{2R}\right) - \frac{\sqrt{3}}{2} D\sqrt{4R^2 - 3D^2} \tag{A.5}$$

Die virtuelle Dichte der mobilen Teilnehmer ist also für $\rho < r$ unverändert und verringert sich im Kreisring $r < \rho < R$ um P_{AP}. Die übergreifende Wahrscheinlichkeitsdichtefunktion der vom Ursprungsort bis zum Zellrand zurückgelegten Entfernung kann nun durch eine einfache Superposition bestimmt werden. Dazu wird die Wahrscheinlichkeitsdichtefunktion der Strecken für den Fall einer Gleichverteilung innerhalb des Radius R mit P_{AP} gewichtet und superpositioniert mit der mit $(1 - P_{AP})$ gewichteten Funktion für den Fall einer Verteilung der Knoten innerhalb des äquivalenten Radius r. Ersteres wurde bereits oben in Gl. (A.3) bestimmt, während hingegen letzteres aus Gl. (A.1) hergeleitet werden kann unter Verwendung der veränderten Wahrscheinlichkeitsdichtefunktion von ρ

$$f''_{\rho}(\rho) = 2\rho/r^2 \quad \text{für} \quad 0 \le \rho \le r \tag{A.6}$$

Mit den gleichen Standardmethoden wie in [HR86] läßt sich dann die zweidimensionale Wahrscheinlichkeitsdichtefunktion von W (Hilfsvariable) und Z''_n angeben als

$$f''_{ZW}(z,w) = \frac{2}{\pi r^2} \frac{z+w}{\sqrt{R^2 - (z+w)^2}} \tag{A.7}$$

$$\text{für} \quad 0 \le |R - z| \le r \qquad \text{und} \quad \frac{R^2 - r^2 - z^2}{2z} \le w \le R - z$$

mit $w = \rho\cos\theta$ wie in Abb. A.1. Die untere Grenze von w läßt sich über den Cosinus-Satz unter der Annahme mit $\rho = r$ herleiten[1]. Schließlich läßt sich nach Eliminierung von w die Wahrscheinlichkeitsfunktion von Z''_n schreiben als

$$\begin{aligned} f''_{Zn}(z) &= \frac{2}{\pi r^2} \int_{\frac{R^2-r^2-z^2}{2z}}^{R-z} \frac{z+w}{\sqrt{R^2-(z+w)^2}}\,\mathrm{d}w \\ &= \frac{2}{\pi r^2}\sqrt{R^2 - \left(\frac{R^2 - r^2 + z^2}{2z}\right)^2} \quad \text{für} \quad 0 \le |R-z| \le r \end{aligned} \tag{A.8}$$

Mittels der zuvor erwähnten Superposition kann nun die Wahrscheinlichkeitsdichtefunktion der Entfernung Z_n, die ein mobiler Knoten vom Ursprungsort bis zum Zellrand zurücklegt angegeben werden als

$$f_{Zn}(z) = \begin{cases} P_{AP}\frac{2}{\pi R^2}\sqrt{R^2 - (\frac{z}{2})^2} + \\ \qquad +(1-P_{AP})\frac{2}{\pi r^2}\sqrt{R^2 - (\frac{R^2-r^2+z^2}{2z})^2} & \text{für} \quad 0 \le |R-z| \le r \\ P_{AP}\frac{2}{\pi R^2}\sqrt{R^2 - (\frac{z}{2})^2} & \text{für} \quad r < |R-z| \le R \end{cases} \tag{A.9}$$

[1] Hinweis: w ist dann negativ.

A.2 Allgemeine Handoverzeit neuer Terminals

Setzt man nun Gl. (A.9) in Gl. (3.14) ein, ergibt sich die Verteilung der allgemeinen Handoverzeiten neuer Knoten als

$$f_{Tn}(t) = P_{AP} \underbrace{\frac{1}{\Delta v}\int_{v_{\min}}^{v_{\max}} v\, f'_{Zn}(vt)\,\mathrm{d}v}_{f'_{Tn}(t)} + (1-P_{AP}) \underbrace{\frac{1}{\Delta v}\int_{v_{\min}}^{v_{\max}} v\, f''_{Zn}(vt)\,\mathrm{d}v}_{f''_{Tn}(t)} \tag{A.10}$$

wiederum mit $\Delta v = v_{\max} - v_{\min}$. Der erste Teil läßt sich analog zu 3.3.1 unter Verwendung von $f'_{Zn}(t)$ wie in Gl. (A.3) lösen:

$$\begin{aligned} f'_{Tn}(t) &= \frac{2}{\pi R^2}\frac{1}{\Delta v}\int_{v_{\min}}^{v_{\max}} v\sqrt{R^2 - \frac{v^2t^2}{4}}\,\mathrm{d}v \quad \text{mit} \quad 0 \le vt \le 2R \\ &= \begin{cases} \frac{8}{3\pi R^2 t^2}\frac{1}{\Delta v}\left(R^2 - \frac{v_{\min}^2 t^2}{4}\right)^{\frac{3}{2}} & \text{für } \frac{2R}{v_{\max}} \le t \le \frac{2R}{v_{\min}} \\ \frac{8}{3\pi R^2 t^2}\frac{1}{\Delta v}\left(\left(R^2 - \frac{v_{\min}^2 t^2}{4}\right)^{\frac{3}{2}} - \left(R^2 - \frac{v_{\max}^2 t^2}{4}\right)^{\frac{3}{2}}\right) & \text{für } 0 \le t \le \frac{2R}{v_{\max}} \end{cases} \end{aligned} \tag{A.11}$$

Da die Lösung des elliptischen Integrals in $f''_{Tn}(t)$ aus Gl. (A.8) auf unhandliche Terme führt, seien hier nur die einzelnen Fälle unter Beibehaltung der Integrale aufgelistet:

$$\begin{aligned} f''_{Tn}(t) &= \frac{2}{\pi r^2}\frac{1}{\Delta v}\int_{v_{\min}}^{v_{\max}} v\sqrt{R^2 - \left(\frac{R^2 - r^2 + v^2t^2}{2vt}\right)^2}\,\mathrm{d}v \quad \text{mit} \quad 0 \le |R - vt| \le r \\ &= \begin{cases} \frac{2}{\pi r^2 t^2}\frac{1}{\Delta v}\int_{v_{\min}t}^{R+r} w\sqrt{R^2 - \left(\frac{R^2-r^2+w^2}{2w}\right)^2}\,\mathrm{d}w & \text{für } \frac{R-r}{v_{\min}} \le t \le \frac{R+r}{v_{\min}} \\ \frac{2}{\pi r^2 t^2}\frac{1}{\Delta v}\int_{R-r}^{R+r} w\sqrt{R^2 - \left(\frac{R^2-r^2+w^2}{2w}\right)^2}\,\mathrm{d}w & \text{für } \frac{R+r}{v_{\max}} \le t \le \frac{R-r}{v_{\min}} \\ \frac{2}{\pi r^2 t^2}\frac{1}{\Delta v}\int_{R-r}^{v_{\max}t} w\sqrt{R^2 - \left(\frac{R^2-r^2+w^2}{2w}\right)^2}\,\mathrm{d}w & \text{für } \frac{R-r}{v_{\max}} \le t \le \frac{R+r}{v_{\max}} \end{cases} \end{aligned} \tag{A.12}$$

Hierbei muß jedoch die Annahme in Gl. (3.19) wie folgt verschärft werden:

$$\frac{R+r}{v_{\max}} \le \frac{R-r}{v_{\min}} \qquad \text{oder} \qquad \frac{v_{\min}}{v_{\max}} \le \frac{R-r}{R+r} \tag{A.13}$$

Abkürzungsverzeichnis

Mit einem * markierte Begriffe sind Elemente der in Kapitel 5 vorgestellten Architektur.

AAA Authentication, Authorization und Accounting

ACK Acknowledge (TCP, R-UDP)

ACKDISC Acknowledge for Disconnect (R-UDP)

AgntAdv Agent Advertisement (Mobile IP)

ALAN Application Level Active Network

AMnet Active Multicasting Network

ANN Active Network Node

ANTS Active Node Transfer System

ATM Asynchronous Transfer Mode

BOOTP Bootstrap Protocol

BS Basisstation

CCM Connection Control and Management

CDMA Code Division Multiple Access

CONN Connect (R-UDP)

CORBA Common Object Request Broker Architecture

CT-Req Context Transfer Request (CXTP)

CTAA Context Transfer Activate Acknowledge (CXTP)

CTAR Context Transfer Activate Request (CXTP)

CTD Context Transfer Data (CXTP)

CTDR Context Transfer Data Reply (CXTP)

CTI Computer Telephony Integration

CXTP Context Transfer Protocol

DHCP Dynamic Host Configuration Protocol

DISC Disconnect (R-UDP)

DNS Domain Name System

EDGE Enhanced Data Rate for GSM Evolution

ESS Extended Service Set (Wireless LAN)

FA Foreign Agent (Mobile IP)

FIN Finish (TCP)

FTP File Transfer Protocol

GPRS General Packet Radio Service

GSM Global System for Mobile Communications (ursprünglich: Groupe Spécial Mobile)

HA Home Agent (Mobile IP)

Hawaii Handoff-Aware Wireless Access Internet Infrastructure

HIP Host Identity Protocol

hot-TCP Handover-Triggered TCP

HR Höheres Routing*

HSDPA High Speed Downlink Packet Access

HTTP Hypertext Transfer Protocol

i3 Internet Indirection Infrastructure

IAPP Inter Access Point Protocol

ICMP Internet Control Message Protocol

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IP Internet Protocol

IPv4, IPv6 Vierte beziehungsweise sechste Version des IP

ISO/OSI International Organization for Standardization / Open System Interconnect

IMS IP Multimedia Subsystem

ISM Industrial, Scientific, and Medical (Frequenzband)

ISUP ISDN User Part Supplementary Services

ITU International Telecommunication Union

LAN Local Area Network

LARA Lancaster Active Router Architecture

LHP Last Hop Protocol (ReSoA)

M-Strg Mobilitätssteuerung*

MAC Medium Access Control, Media Access Control

MAP Mobility Anchor Point (Mobile IPv6)

ME Mobilitätserkennung*

Mgmt Management*

MIP Mobile IP

Mombasa Mobility Support – A Multicast-Based Approach

MPA Mobility Proxy Agent

MR Mobilitätsrouting*

MTU Maximum Transmission Unit

NACK Negative Bestätigung

NAPT Network Address and Port Translation

NAT Network Address Translation

PFAN Previous Foreign Agent Notification

PIRST-ON Programmable, Intermediate, Resilient, Self-Configuring, Transparent Overlay Network

PLAN Programming Language for Active Networks

RCANE Resource Controlled Active Network Environment

RED Random Early Detection

RegReply Registration Reply (Mobile IP)

RegReq Registration Request (Mobile IP)

ReSoA Remote Socket Architecture

RMS Resilient Mobile Socket

RO Routenoptimierung (Mobile IP)

RoHC Robust Header Compression

RPC Remote Procedure Call

RSVP Resource ReSerVation Protocol

RTP Real-Time Transport Protocol

R-UDP Reliable UDP

SCTP Stream Control Transmission Protocol

SIP Session Initiation Protocol

SLP Service Location Protocol

SOAP Simple Object Access Protocol

SS7 Signalling System No. 7

SSID Service Set Identifier (Wireless LAN)

STP Self-spreading Transport Protocols

Strg Steuerung*

SYN Synchronization (TCP)

TCP Transmission Control Protocol

TTL Time to Live

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

VDE Verband der Elektrotechnik Elektronik Informationstechnik

VERA Virtuelle Router Architektur

VPRN Virtual Private Routed Network

WiMax Worldwide Interoperability for Microwave Access

WLAN Wireless LAN

WSDL Web Service Definition Language

WWW World Wide Web

XML Extensible Markup Language

Abbildungsverzeichnis

Tabellenverzeichnis

Stichwortverzeichnis

Literaturverzeichnis

Literaturangaben, deren Verweis im Fettdruck erscheint, sind Vorveröffentlichungen des Verfassers die als Haupt- oder Mitautor erstellt wurden.

[3GP04] 3RD GENERATION PARTNERSHIP PROJECT: *High Speed Downlink Packet Access (HSDPA)*. 3GPP-TS25.308, Dezember 2004.

[3GP05] 3RD GENERATION PARTNERSHIP PROJECT: *All-IP Network (AIPN) Feasibility Study (Release 7)*. 3GPP-TS22.978, Juni 2005.

[3GP06] 3RD GENERATION PARTNERSHIP PROJECT: *IP Multimedia Subsystem (IMS), Stage 2 (Release 7)*. 3GPP-TS23.228, März 2006.

[AAH+98] ALEXANDER, D. S., W. A. ARBAUGH, M. W. HICKS et al.: *The SwitchWare Active Network Architecture*. IEEE Network, 12(3):29–36, Mai 1998.

[AMK+01] ALEXANDER, D. S., P. B. MENAGE, A. D. KEROMYTIS et al.: *The Price of Safety in An Active Network*. Journal of Communications and Networks (JCN), 3(1):4–18, März 2001.

[Apa] APACHE SOFTWARE FOUNDATION: *Apache HTTP Server Project*. `http://www.apache.org/`.

[BB97] BAKRE, A. V. und B. R. BADRINATH: *Implementation and Performance Evaluation of Indirect TCP*. IEEE Transactions on Computers, 46(3):260–278, 1997.

[BBD+01] BORMANN, C., C. BURMEISTER, M. DEGERMARK et al.: *RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed* . RFC 3095 (Standardisierungsvorschlag), IETF, Juli 2001. Aktualisiert durch RFC 3759.

[BBR00] BERSON, S., B. BRADEN und L. RICCIULLI: *Introduction to the ABone*. University of Southern California, Information Sciences Institute (ISI), `http://www.isi.edu/abone/`, Juni 2000.

[Bet01] BETTSTETTER, C.: *Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects*. ACM SIGMOBILE Mobile Computing and Communications Review, 5(3):55–66, Juli 2001.

[BHPC04] BETTSTETTER, C., H. HARTENSTEIN und X. PÉREZ-COSTA: *Stochastic Properties of the Random Waypoint Mobility Model*. ACM/Kluwer Wireless Networks, 10(5):555–567, September 2004.

[Bic05] BICHLER, S.: *Protokollheader-Komprimierung für das mobile Internet*. Interdisziplinäres Projekt, Technische Universität München, Lehrstuhl für Kommunikationsnetze (LKN), April 2005.

[BLH+98] BISWAS, J., A. A. LAZAR, J.-F. HUARD et al.: *The IEEE P1520 Standards Initiative for Programmable Network Interfaces*. IEEE Communications Magazine, 36(10):64–70, Oktober 1998.

[Boc90] BOCKER, P.: *ISDN - Das diensteintegrierende digitale Nachrichtennetz*. Springer-Verlag Berlin Heidelberg, Berlin Heidelberg, 1990.

[BPSK97] BALAKRISHNAN, H., V. N. PADMANABHAN, S. SESHAN und R. H. KATZ: *A Comparison of Mechanisms for Improving TCP Performance over Wireless Links*. IEEE/ACM Transactions on Networking, 5(6):756–769, Dezember 1997.

[BPT96] BHAGWAT, P., C. PERKINS und S. TRIPATHI: *Network Layer Mobility: An Architecture and Survey*. IEEE Personal Communications, 3(3):54–64, Juni 1996.

[BS97] BROWN, K. und S. SINGH: *M-TCP: TCP for Mobile Cellular Networks*. ACM SIGCOMM Computer Communication Review, 27(5), April 1997.

[**BT02**] BACHMEIR, C. und P. TABERY: *PIRST-ONs: A Service Architecture for Embedding and Leveraging Active and Programmable Network Technology*. In: *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Oktober 2002.

[**BTS+03**] BACHMEIR, C., P. TABERY, E. SFEIR et al.: *HArPooN: A Scalable, High Performance, Fault Tolerant Programmable Router Architecture*. In: *Poster Session of the International Working Conference on Active Networks (IWAN)*, Dezember 2003.

[BVE99] BETTSTETTER, C., H.-J. VÖGEL und J. EBERSPÄCHER: *GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface*. IEEE Communications Surveys & Tutorials, 2(3), Third Quarter 1999.

[BZB+97] BRADEN, R., L. ZHANG, S. BERSON et al.: *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*. RFC 2205 (Status: Standardisierungsvorschlag), IETF, September 1997. Aktualisiert durch RFCs 2750, 3936.

[CBZS98] CALVERT, K., S. BHATTACHARJEE, E. ZEGURA und J. STERBENZ: *Directions in Active Networks*. IEEE Communications Magazine, 36(10):72–78, Oktober 1998.

[CCR+03] CHUN, B., D. CULLER, T. ROSCOE et al.: *PlanetLab: An Overlay Testbed for Broad-Coverage Services*. ACM SIGCOMM Computer Communication Review, 33(3):3–12, Juli 2003.

[CD01] COOPER, I. und J. DILLEY: *Known HTTP Proxy/Caching Problems*. RFC 3143 (Informatorisch), IETF, Juni 2001.

[CDK+99] CAMPBELL, A. T., H. G. DE MEER, M. E. KOUNAVIS et al.: *A Survey of Programmable Networks*. ACM SIGCOMM Computer Communication Review, 29(2), April 1999.

[CG85] CROFT, W. und J. GILMORE: *Bootstrap Protocol*. RFC 951 (Status: Vorläufiger Standard), IETF, September 1985. Aktualisiert durch RFCs 1395, 1497, 1532, 1542.

[Cha05] CHADD, A.: *Squid Web Proxy Cache*, 2000-2005. `http://www.squid-cache.org/`.

[Che95] CHEEK, M.: *Computer Telephony Integration*. IEEE Computer, 28(6):8–9, Juni 1995.

[Chi06] CHITTARO, L.: *Visualizing Information on Mobile Devices*. IEEE Computer, 39(3), März 2006.

[Chr03] CHRISTENSEN, C. M.: *The Innovator's Dilemma*. HarperCollins, New York, 2003.

[CKV01] CAMPBELL, A. T., M. E. KOUNAVIS und J. B. VICENTE: *Informatics - 10 Years Back, 10 Years Ahead*, Kapitel Programmable Networks. Springer-Verlag, Berlin Heidelberg, 2001.

[CV93] CAMERON, E. J. und H. VELTHUIJSEN: *Feature Interactions in Telecommunications Systems*. IEEE Communications Magazine, 31(8):18–23, August 1993.

[DDPP00] DECASPER, D., Z. DITTIA, G. PARULKAR und B. PLATTNER: *Router Plugins: A Software Architecture for Next Generation Routers*. IEEE/ACM Transactions on Networking, 8(1):2–15, Februar 2000.

[Dro97] DROMS, R.: *Dynamic Host Configuration Protocol*. RFC 2131 (Status: Vorläufiger Standard), IETF, März 1997. Aktualisiert durch RFC 3396.

[DST04] DUNGS, D., H.-P. SCHWEFEL und P. TABERY: *Coupling Mobility Support and TCP Performance Enhancement in Heterogeneous Wireless Networks: An Experimental Evaluation*. In: *International Symposium on Wireless Personal Multimedia Communications (WPMC)*, September 2004.

[Ebe02] EBERSPÄCHER, J.: *Handbuch für die Telekommunikation*, Kapitel Vermittlung und Mobilität. Springer-Verlag, Berlin Heidelberg, 2002.

[Ebe06] EBERSPÄCHER, J.: *Trends in Telecommunication Networking*. In: BRÜGGE, B. und H.-G. HEGERING (Herausgeber): *Managing Development and Application of Digital Technologies*, Seiten 50–56. Springer-Verlag, Juni 2006.

[EFH+96a] EBERSPÄCHER, J., N. FISCHER, P. HÄRLE et al.: *Architekturen und Verfahren der Vermittlungstechnik*. ITG Empfehlung 5.2-01. VDE-Verlag, Berlin, Offenbach, Dezember 1996.

[EFH+96b] EBERSPÄCHER, J., N. FISCHER, P. HÄRLE et al.: *Systeme der Vermittlungstechnik*. ITG Empfehlung 5.2-02. VDE-Verlag, Berlin, Offenbach, Dezember 1996.

[EM05] EL MALKI, K.: *Low Latency Handoffs in Mobile IPv4, <draft-ietf-mobileip-lowlatency-handoffs-v4-11>*. Internet Draft, IETF, Oktober 2005. Herausgabe als RFC in Vorbereitung.

[Eri94] ERIKSSON, H.: *MBONE: The Multicast Backbone*. Communications of the ACM, 37(8):54–60, August 1994.

[EVB01] EBERSPÄCHER, J., H.-J. VÖGEL und C. BETTSTETTER: *GSM Global System for Mobile Communication*. Teubner, Stuttgart Leipzig Wiesbaden, 2001.

[FA04] FU, S. und M. ATIQUZZAMAN: *SCTP: State of the Art in Research, Products, and Technical Challenges*. IEEE Communications Magazine, 42(4):64–76, April 2004.

[FAFH04] FONG, B., N. ANSARI, A. C. M. FONG und G. Y. HONG: *On the Scalability of Fixed Broadband Wireless Access Network Deployment*. IEEE Communications Magazine, 42(9):S12–S18, September 2004.

[Fes03] FESTAG, A.: *Mobility Support in IP Cellular Networks–A Multicast-Based Approach*. Doktorarbeit, Technische Universität Berlin, 2003.

[FG99] FRY, M. und A. GHOSH: *Application Level Active Networking*. Computer Networks, 31(7):655–667, April 1999.

[FG01] FIKOURAS, N. A. und C. GÖRG: *Performance Comparison of Hinted- and Advertisement-based Movement Detection Methods for Mobile IP Hand-offs*. Computer Networks, 37(1):55–62, September 2001.

[FHG04] FLOYD, S., T. HENDERSON und A. GURTOV: *The NewReno Modification to TCP's Fast Recovery Algorithm*. RFC 3782 (Status: Standardisierungsvorschlag), IETF, April 2004.

[FHSZ02] FUHRMANN, T., T. HARBAUM, M. SCHÖLLER und M. ZITTERBART: *AMnet 2.0: An Improved Architecture for Programmable Networks*. In: *International Working Conference on Active Networks (IWAN)*, Dezember 2002.

[FMM+99] FORSBERG, D., J. T. MALINEN, J. K. MALINEN et al.: *Distributing Mobility Agents Hierarchically under Frequent Location Updates*. In: *IEEE International Workshop on Mobile Multimedia Communications (MoMuC)*, November 1999.

[FMMP00] FLOYD, S., J. MAHDAVI, M. MATHIS und M. PODOLSKY: *An Extension to the Selective Acknowledgement (SACK) Option for TCP*. RFC 2883 (Status: Standardisierungsvorschlag), IETF, Juli 2000.

[FMMW99] FORSBERG, D., J. T. MALINEN, J. K. MALINEN und T. WECKSTRÖM: *Dynamics HUT Mobile IP*. Helsinki University of Technology, Oktober 1999. `http://www.cs.hut.fi/Research/Dynamics/`.

[Fri96] FRITSCHE, N.: *Vermittlungsarchitektur mit getrennter Ruf- und Leistungsmerkmalsteuerung*. Doktorarbeit, Technische Universität München, 1996.

[FS00] FERGUSON, P. und D. SENIE: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. RFC 2827 (Status: Bewährte Vorgehensweise), IETF, Mai 2000. Aktualisiert durch RFC 3704.

[FV93] FLOYD, S. und VAN JACOBSON: *Random Early Detection Gateways for Congestion Avoidance*. IEEE/ACM Transactions on Networking, 1(4):397–413, August 1993.

[FWW02] FESTAG, A., L. WESTERHOFF und A. WOLISZ: *The MOMBASA Software Environment - A Toolkit for Performance Evaluation of Multicast-Based Mobility Support*. In: *International Conference on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation (PERFORMANCE TOOLS)*, April 2002.

[GKS02] GOKHALE, A., B. KUMAR und A. SAHUGUET: *Reinventing the Wheel? CORBA vs. Web Services*. In: *International World Wide Web Conference*, Mai 2002.

[GLH+00] GLEESON, B., A. LIN, J. HEINANEN et al.: *A Framework for IP Based Virtual Private Networks*. RFC 2764 (Status: Informatorisch), IETF, Februar 2000.

[HB04] HAAS, H. und A. BROWN: *Web Services Glossary*, Februar 2004. `http://www.w3.org/TR/ws-gloss/`.

[Hen06] HENDERSON, T.: *End-Host Mobility and Multihoming with the Host Identity Protocol, <draft-ietf-hip-mm-03>*. Internet Draft, IETF, Februar 2006.

[Her88] HERTZ, H.: *Über die Ausbreitungsgeschwindigkeit der elektrodynamischen Wirkungen*. In: *Sitzungsberichte der Kgl. Preuss. Akademie der Wissenschaften zu Berlin, Physikal.-Math. Classe*, 1888.

[HGJ+99] HAAS, Z. J., M. GERLA, D. B. JOHNSON et al.: *Guest Editorial Wireless Ad Hoc Networks*. IEEE Journal on Selected Areas in Communications, 17(8):1329–1332, August 1999.

[HKM+98] HICKS, M., P. KAKKAR, J. T. MOORE et al.: *PLAN: A Packet Language For Active Networks*. In: *ACM SIGPLAN International Conference on Functional Programming (ICFP)*, September 1998.

[HMA+99] HICKS, M., J. MOORE, D. ALEXANDER et al.: *PLANet: An Active Internetwork*. In: *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, März 1999.

[HR86] HONG, D. und S. S. RAPPAPORT: *Traffic Model and Performance Analysis for Cellular Mobile Radio Telephone Systems with Prioritized and Nonprioritized Handoff Procedures*. IEEE Transactions on Vehicular Technology, 35(3):77–92, August 1986.

[IEE99a] IEEE STANDARDS BOARD: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (ANSI/IEEE 802.11)*, 1999.

[IEE99b] IEEE STANDARDS BOARD: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band (IEEE Std 802.11a)*, 1999.

[IEE99c] IEEE STANDARDS BOARD: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band (IEEE Std 802.11b)*, 1999.

[IEE03] IEEE STANDARDS BOARD: *IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation (IEEE Std 802.11F)*, 2003.

[Int93a] INTERNATIONAL TELECOMMUNICATION UNION: *Integrated Services Digital Network (ISDN) - Vocabulary of Terms for ISDNs, ITU-T Recommendation I.112*, März 1993.

[Int93b] INTERNATIONAL TELECOMMUNICATION UNION: *Open Systems Interconnection - Basic Reference Model, ITU-T Recommendation X.200*, März 1993.

[Int93c] INTERNATIONAL TELECOMMUNICATION UNION: *Specifications of Signalling System No.7, ITU-T Recommendation Q.700*, März 1993.

[Int99] INTERNATIONAL TELECOMMUNICATION UNION: *Specifications of Signalling System No.7 – ISDN User Part, ITU-T Recommendation Q.761*, Dezember 1999.

[IS95] IVANOV, K. und G. SPRING: *Mobile Speed Sensitive Handover in a Mixed Cell Environment*. In: *IEEE Vehicular Technology Conference (VTC)*, Juli 1995.

[Jac88] JACOBSON, V.: *Congestion Avoidance and Control*. In: *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 1988.

[JB00] JUGL, E. und H. BOCHE: *Analysis of Analytical Mobility Models with Respect to the Applicability for Handover Modeling and to the Estimation of Signaling Cost*. In: *International Conference on Mobile Computing and Networking (MobiCom)*, August 2000.

[JPA04] JOHNSON, D., C. PERKINS und J. ARKKO: *Mobility Support in IPv6*. RFC 3775 (Status: Standardisierungsvorschlag), IETF, Juni 2004.

[KAL+01] KAARANEN, H., A. AHTIANEN, L. LAITINEN et al.: *UMTS Networks*. John Wiley & Sons, LTD, Chichester, 2001.

[KCD+00] KELLER, R., S. CHOI, M. DASEN et al.: *An Active Router Architecture for Multicast Video Distribution*. In: *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, März 2000.

[Kem02] KEMPF, J.: *Problem Description: Reasons for Performing Context Transfers Between Nodes in an IP Access Network*. RFC 3374 (Status: Informatorisch), IETF, September 2002.

[KL04] KOSSMANN, D. und F. LEYMANN: *Web-Services*. Informatik-Spektrum, 27(2):117–128, April 2004.

[KLR+06a] KEMPF, J., K. LEUNG, P. ROBERTS et al.: *Problem Statement for IP Local Mobility, <draft-ietf-netlmm-nohost-ps-01>*. Internet Draft, IETF, April 2006.

[KLR+06b] KEMPF, J., K. LEUNG, P. ROBERTS et al.: *Requirements and Gap Analysis for IP Local Mobility, <draft-ietf-netlmm-nohost-req-00>*. Internet Draft, IETF, Februar 2006.

[KMC+00] KOHLER, E., R. MORRIS, B. CHEN et al.: *The Click Modular Router*. ACM Transactions on Computer Systems, 18(3):263–297, August 2000.

[Koo05] KOODLI, R.: *Fast Handovers for Mobile IPv6*. RFC 4068 (Status: Experimentell), IETF, Juli 2005.

[KP01a] KARLIN, S. und L. PETERSON: *VERA: An Extensible Router Architecture*. In: *IEEE Conference on Open Architectures and Network Programming (OPENARCH)*, April 2001.

[KP01b] KOODLI, R. und C. PERKINS: *Fast Handovers and Context Transfers in Mobile Networks*. ACM SIGCOMM Computer Communication Review, 31(5):37–47, Oktober 2001.

[KP06] KRISTIANSSON, J. und P. PARNESON: *An Application-Layer Approach to Seamless Mobile Multimedia Communication*. IEEE eTransactions on Network and Service Management (eTNSM), 2(1):33–42, Januar 2006.

[KSM00] KELLERER, W., P. STIES und P. MORITZ: *Service Interactions beyond IN: The new Challenge for Multimedia and Convergence*. In: *International Conference on Intelligence in Networks (ICIN)*, Januar 2000.

[Lag97] LAGRANGE, X.: *Multitier Cell Design*. IEEE Communications Magazine, 35(8):60–64, August 1997.

[LCC+03] LEINER, B. M., V. G. CERF, D. D. CLARK et al.: *A Brief History of the Internet*, Dezember 2003. `http://www.isoc.org/internet/history/brief.shtml`.

[LDV+99] LIN, P., S. DENAZIS, J. VICENTE et al.: *Programming Interfaces for IP Routers and Switches, an Architectural Framework Document*. Standardisierungsentwurf, Working Group for IEEE P1520, Januar 1999.

[LDY06] LEUNG, K., G. DOMMETY und P. YEGANI: *Mobility Management using Proxy Mobile IPv4, <draft-leung-mip4-proxy-mode-00>*. Internet Draft, IETF, Februar 2006.

[LGL+96] LEECH, M., M. GANIS, Y. LEE et al.: *SOCKS Protocol Version 5*. RFC 1928 (Status: Standardisierungsvorschlag), IETF, März 1996.

[LGT98] LEHMAN, L., S. GARLAND und D. TENNENHOUSE: *Active Reliable Multicast*. In: *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, März 1998.

[Lin06] LINZ, S.: *Preise in Deutschland 2006*. Statistisches Bundesamt, Wiesbaden, Mai 2006.

[LNPK05] LOUGHNEY, J., M. NAKHJIRI, C. PERKINS und R. KOODLI: *Context Transfer Protocol (CXTP)*. RFC 4067 (Status: Experimentell), IETF, Juli 2005.

[LR96] LIBERTI, J. C. und T. S. RAPPAPORT: *A Geometrically Based Model for Line-of-Sight Multipath Radio Channels*. In: *IEEE Vehicular Technology Conference (VTC)*, Band 2, Seiten 844–848, Mai 1996.

[MB98] MALTZ, D. A. und P. BHAGWAT: *MSOCKS: An Architecture for Transport Layer Mobility*. In: *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, März 1998.

[MBZC00] MERUGU, S., S. BHATTACHARJEE, E. ZEGURA und K. CALVERT: *Bowman: A Node OS for Active Networks*. In: *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, März 2000.

[McC05] MCCANN, P.: *Mobile IPv6 Fast Handovers for 802.11 Networks*. RFC 4260 (Informatorisch), IETF, November 2005.

[MD90] MOGUL, J. und S. DEERING: *Path MTU discovery*. RFC 1191 (Status: Vorläufiger Standard), IETF, November 1990.

[Mit95] MITOLA, J.: *The Software Radio Architecture*. IEEE Communications Magazine, 33(5):26–38, Mai 1995.

[Mit03] MITRA, N.: *SOAP Version 1.2 Part 0: Primer*. Technischer Bericht, World Wide Web Consortium (W3C), Juni 2003.

[MK04] MANNER, J. und M. KOJO: *Mobility Related Terminology*. RFC 3753 (Informatorisch), IETF, Juni 2004.

[MN06] MOSKOWITZ, R. und P. NIKANDER: *Host Identity Protocol (HIP) Architecture*. RFC 4423 (Status: Informatorisch), IETF, Mai 2006.

[MNJH06] MOSKOWITZ, R., P. NIKANDER, P. JOKELA und T. HENDERSON: *Host Identity Protocol, <draft-ietf-hip-base-05>*. Internet Draft, IETF, März 2006.

[Moc87] MOCKAPETRIS, P.: *Domain Names - Implementation and Specification*. RFC 1035 (Status: Standard), IETF, November 1987. Aktualisiert durch RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035.

[Mon01] MONTENEGRO, G.: *Reverse Tunneling for Mobile IP, revised*. RFC 3024 (Status: Standardisierungsvorschlag), IETF, Januar 2001.

[MRS+99] MANIATIS, P., M. ROUSSOPOULOS, E. SWIERK et al.: *The Mobile People Architecture*. ACM Mobile Computing and Communications Review (MC2R), 3(3):36–42, Juli 1999.

[MSA03] MISHRA, A., M. SHIN und W. ARBAUGH: *An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process*. ACM SIGCOMM Computer Communication Review, 33(2):93–102, April 2003.

[MSKC04] MCKINLEY, P. K., S. M. SADJADI, E. P. KASTEN und B. H. C. CHENG: *Composing Adaptive Software*. IEEE Computer, 37(7):56–64, Juli 2004.

[MT02] MORAND, L. und S. TESSIER: *Global Mobility Approach with Mobile IP in All IP Networks*. In: *IEEE International Conference on Communications (ICC)*, April 2002.

[NAA+05] NIKANDER, P., J. ARKKO, T. AURA et al.: *Mobile IP Version 6 Route Optimization Security Design Background*. RFC 4225 (Status: Informatorisch), IETF, Dezember 2005.

[Net04] NETGEAR INC.: *Product Data Sheet of WAG511 ProSafe Dual Band Wireless PC Card*, 2004.

[Neu02] NEUSS, W.: *Handbuch für die Telekommunikation*, Kapitel Ergänzende Dienste auf Basis der Vermittlungssysteme und Server für Sprachdienste (Call/Feature Server). Springer-Verlag, Berlin Heidelberg, 2002.

[Obj04a] OBJECT MANAGEMENT GROUP, INC.: *Common Object Request Broker Architecture: Core Specification, Version 3.0.3*, März 2004.

[Obj04b] OBJECT MANAGEMENT GROUP, INC.: *Wireless Access and Terminal Mobility in CORBA*, April 2004.

[OJ03] OHRTMAN JR., F. D.: *Softswitch*. McGraw-Hill, New York, 2003.

[OY02] ONG, L. und J. YOAKUM: *An Introduction to the Stream Control Transmission Protocol (SCTP)*. RFC 3286 (Status: Informatorisch), IETF, Mai 2002.

[PA00] PAXSON, V. und M. ALLMAN: *Computing TCP's Retransmission Timer*. RFC 2988 (Status: Standardisierungsvorschlag), IETF, November 2000.

[PD00] PATEL, G. und S. DENNETT: *The 3GPP and 3GPP2 Movements toward an All-IP Mobile Network*. IEEE Personal Communications, 7(4):62–64, August 2000.

[Per98] PERKINS, C.: *Mobile IP: Design Principles and Practices*. Addison-Wesley, Reading, 1998.

[Per00] PERLMAN, R.: *Interconnections*. Addison-Wesley, Reading, 2000.

[Per02] PERKINS, C.: *IP Mobility Support for IPv4*. RFC 3344 (Status: Standardisierungsvorschlag), IETF, August 2002.

[PJ98] PERKINS, C. und D. B. JOHNSON: *Route Optimization for Mobile IP*. Cluster Computing, 1(2):161–176, September 1998.

[Pos81] POSTEL, J.: *Internet Protocol*. RFC 791 (Status: Standard), IETF, September 1981. Aktualisiert durch RFC 1349.

[PR85] POSTEL, J. und J. REYNOLDS: *File Transfer Protocol*. RFC 959 (Status: Standard), IETF, Oktober 1985. Aktualisiert durch RFCs 2228, 2640, 2773.

[Pso99] PSOUNIS, K.: *Active Networks: Applications, Security, Safety, and Architectures*. IEEE Communications Surveys & Tutorials, 2(1), Januar 1999.

[PWW+03] PATEL, P., A. WHITAKER, D. WETHERALL et al.: *Upgrading Transport Protocols using Untrusted Mobile Code*. In: *ACM Symposium on Operating Systems Principles*, Oktober 2003.

[RAB06] REZAHFAR, R., P. AGASHE und P. BENDER: *Macro-Mobility Management in EVDO*. IEEE Communications Magazine, 44(2):103–110, Februar 2006.

[Ric00] RICHARD, G.G., I.: *Service Advertisement and Discovery: Enabling Universal Device Cooperation*. IEEE Internet Computing, 4(5):18–26, September 2000.

[RLTV99] RAMJEE, R., T. LA PORTA, S. THUEL und K. VARADHAN: *HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks*. In: *International Conference on Network Protocols (ICNP)*, Oktober 1999.

[RSC+02] ROSENBERG, J., H. SCHULZRINNE, G. CAMARILLO et al.: *SIP: Session Initiation Protocol*. RFC 3261 (Status: Standardisierungsvorschlag), IETF, Juni 2002. Aktualisiert durch RFCs 3265, 3853.

[RT06] RIEGEL, M. und M. TÜXEN: *Mobile SCTP, <draft-riegel-tuexen-mobile-sctp-06>*. Internet Draft, IETF, März 2006.

[RVS+02] RAMJEE, R., K. VARADHAN, L. SALGARELLI et al.: *HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks*. IEEE/ACM Transactions on Networking, 10(3):396–410, Juni 2002.

[SAZ+04] STOICA, I., D. ADKINS, S. ZHUANG et al.: *Internet Indirection Infrastructure*. IEEE/ACM Transactions on Networking, 12(2):205–218, April 2004.

[SB00] SNOEREN, A. C. und H. BALAKRISHNAN: *An End-to-End Approach to Host Mobility*. In: *International Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[Sch02] SCHMIDT, S.: *Migration mobiler Dienste im Internet*. Diplomarbeit, Lehrstuhl für Kommunikationsnetze, Technische Universität München, 2002.

[Sch04] SCHLÄGER, M.: *The Remote Socket Architecture: A Proxy Based Solution for TCP over Wireless*. Doktorarbeit, Technische Universität Berlin, 2004.

[SCMB05] SOLIMAN, H., C. CASTELLUCCIA, K. E. MALKI und L. BELLIER: *Hierarchical Mobile IPv6 Mobility Management (HMIPv6)*. RFC 4140 (Status: Experimentell), IETF, August 2005.

[SE01] SRISURESH, P. und K. EGEVANG: *Traditional IP Network Address Translator (Traditional NAT)*. RFC 3022 (Status: Informatorisch), IETF, Januar 2001.

[Sen02] SENIE, D.: *Network Address Translator (NAT)-Friendly Application Design Guidelines*. RFC 3235 (Status: Informatorisch), IETF, Januar 2002.

[SFRS04] SHIN, S., A. G. FORTE, A. S. RAWAT und H. SCHULZRINNE: *Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs*. In: *International Workshop on Mobility Management and Wireless Access (MobiWac)*, September 2004.

[SFSS01] SCHMID, S., J. FINNEY, A. SCOTT und W. SHEPHERD: *Component-based Active Network Architecture*. In: *IEEE Symposium on Computers and Communications*, Seiten 114–121, Juli 2001.

[SK98] STEMM, M. und R. H. KATZ: *Vertical Handoffs in Wireless Overlay Networks*. Mobile Networks and Applications, 3(4):335–350, Dezember 1998.

[SKB+01] SANDERS, M., M. KEATON, S. BHATTACHARJEE et al.: *Active Reliable Multicast on CANEs: A Case Study*. In: *IEEE Conference on Open Architectures and Network Programming (OPENARCH)*, April 2001.

[SPG+03] SIMOENS, S., P. PELLATI, J. GOSTEAU et al.: *The Evolution of 5GHz WLAN toward Higher Throughputs*. IEEE Wireless Communications, 10(6):6–13, Dezember 2003.

[SRBW01] SCHLÄGER, M., B. RATHKE, S. BODENSTEIN und A. WOLISZ: *Advocating a Remote Socket Architecture for Internet Access using Wireless LANs*. Mobile Networks and Applications (Special Issue on Wireless Internet and Intranet Access), 6(1):23–42, Januar 2001.

[Sri95] SRINIVASAN, R.: *RPC: Remote Procedure Call Protocol Specification Version 2*. RFC 1831 (Status: Standardisierungsvorschlag), IETF, August 1995.

[SRX+06] STEWART, R., M. RAMALHO, Q. XIE et al.: *Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, <draft-ietf-tsvwg-addip-sctp-14>*. Internet Draft, IETF, März 2006.

[SS02] SCHWABE, T. und J. SCHÜLER: *A Comparison of the Performance of four TCP Versions during mobile Handoff*. In: *International Conference on Mobile and Wireless Communication Networks (MWCN)*, September 2002.

[SW00] SCHULZRINNE, H. und E. WEDLUND: *Application-Layer Mobility using SIP*. Mobile Computing and Communications Review, 4(3):47–57, Juli 2000.

[TB02] TABERY, P. und C. BACHMEIR: *Advanced Network Services using Programmable Networks*. In: *IFIP Workshop and EUNICE Summer School on Adaptable Networks and Teleservices*, September 2002.

[TBH05] TABERY, P., S. BICHLER und C. HARTMANN: *Enhanced Modeling of Wireless LAN Inter-Handoff Times*. In: *International Symposium on Wireless Personal Multimedia Communications (WPMC)*, September 2005.

[TBL03] TABERY, P., C. BACHMEIR und X. LI: *Optimizing Mobile Network Datagram Routing through Programmable Proxying*. In: *International Conference on Mobile and Wireless Communications Networks (MWCN)*, Oktober 2003.

[TBW04] TABERY, P., C. BACHMEIR und N. WIMMER: *Introducing Handoff-Triggered TCP (hot-TCP) for Throughput-Optimized Mobility Support*. In: *World Multi-Conference on Systemics, Cybernetics and Informatics (SCI)*, Juli 2004.

[TH06] TABERY, P. und C. HARTMANN: *To Transfer or Not to Transfer—Service Handovers in Wireless LAN Systems with Imperfect Movement Estimation*. International Journal of Electronics and Communications (AEÜ), 60(1), Januar 2006.

[TS03] TABERY, P. und C. SCHWINGENSCHLÖGL: *Remote Header Compression for Optimizing Roaming Users' Wireless Link Performance*. In: *International Conference on Wireless Networks (ICWN)*, Juni 2003.

[TSS+97] TENNENHOUSE, D., J. SMITH, W. SINCOSKIE et al.: *A Survey of Active Network Research*. IEEE Communications Magazine, 35(1):80–86, Januar 1997.

[TSSB04] TABERY, P., C. SCHWINGENSCHLÖGL, D. SCHMIDT und C. BACHMEIR: *Handoff-Triggered TCP (hot-TCP): Performance and Fairness Evaluation*. In: *IEEE Vehicular Technology Conference (VTC)*, September 2004.

[Val99] VALKO, A. G.: *Cellular IP - A New Approach to Internet Host Mobility*. ACM SIGCOMM Computer Communication Review, 29(1):50–65, Januar 1999.

[vdHKvM98] VAN DEN HOUT, K., A. KOOPAL und R. VAN MOOK: *Management of IP numbers by peg-dhcp*. RFC 2322 (Status: Informatorisch), IETF, April 1998.

[VGPK97] VEIZADES, J., E. GUTTMAN, C. PERKINS und S. KAPLAN: *Service Location Protocol*. RFC 2165 (Status: Standardisierungsvorschlag), IETF, Juni 1997. Aktualisiert durch RFCs 2608, 2609.

[Vin97] VINOSKI, S.: *CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments*. IEEE Communications Magazine, 35(2):46–55, Februar 1997.

[VTRB97] VIXIE, P., S. THOMSON, Y. REKHTER und J. BOUND: *Dynamic Updates in the Domain Name System (DNS UPDATE)*. RFC 2136 (Status: Standardisierungsvorschlag), IETF, April 1997. Aktualisiert durch RFCs 3007, 4033, 4034, 4035.

[Web99] WEBSTER, J. G. (Herausgeber): *Wiley Encyclopedia of Electrical and Electronics Engineering*, Band 21, Kapitel Submarine Telegraphy, Seiten 470–480. John Wiley & Sons, Inc., New York, 1999.

[Wel05] WELTE, H.: *The netfilter.org Project*, 1999-2005. `http://www.netfilter.org/`.

[WGT98] WETHERALL, D., J. GUTTAG und D. TENNENHOUSE: *ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols*. In: *IEEE Conference on Open Architectures and Network Programming (OPENARCH)*, April 1998.

[WLG98] WETHERALL, D., D. LEGEDZA und J. GUTTAG: *Introducing New Internet Services: Why and How*. IEEE Network, 12(3):12–19, Mai 1998.

[WRSW04] WESTERHOFF, L., S. REINHARDT, G. SCHÄFER und A. WOLISZ: *Security Analysis and Concept for the Multicast-based Handover Support Architecture MOMBASA*. In: *IEEE Global Telecommunications Conference (GLOBECOM)*, Band 4, Seiten 2201–2207, November 2004.

[XB02] XI, J. und C. BETTSTETTER: *Wireless Multi-Hop Internet Access: Gateway Discovery, Routing, and Addressing*. In: *Conference on Third Generation Wireless and Beyond (3Gwireless)*, Mai 2002.

[XG93] XIE, H. und D. J. GOODMAN: *Mobility Models and Biased Sampling Problem*. In: *International Conference on Universal Personal Communications (ICUPC)*, Oktober 1993.

[XKWM02] XING, W., H. KARL, A. WOLISZ und H. MÜLLER: *M-SCTP: Design and Prototypical Implementation of an End-to-End Mobility Concept*. In: *International Workshop The Internet Challenge: Technology and Applications*, Oktober 2002.

[YIHK02] YOKOTA, H., A. IDOUE, T. HASEGAWA und T. KATO: *Link Layer Assisted Mobile IP Fast Handoff Method over Wireless LAN Networks*. In: *International Conference on Mobile Computing and Networking (MobiCom)*, September 2002.

[Zav93] ZAVE, P.: *Feature Interactions and Formal Specifications in Telecommunications*. IEEE Computer, 26(8):20–28, 30, August 1993.

[ZCB01] ZHAO, X., C. CASTELLUCCIA und M. BAKER: *Flexible Network Support for Mobile Hosts*. Mobile Networks and Applications, 6(2):137–149, März 2001.

[**ZTB03**] ZÜNDT, M., P. TABERY und C. BACHMEIR: *Seamless Handoff in Community Based and Location Aware Heterogenous Wireless Networks*. In: *IEEE International Workshop on Mobile Multimedia Communications (MoMuC)*, Oktober 2003.