Universität Karlsruhe (TH)
Institut für Telematik

TeleMatics Technical Reports

Mobiles Internet
Seminar WS05/06

Thomas Gamer, Stephan Krause, Tobias Küfner, Christian Vogt,
Prof. Dr. Martina Zitterbart
{gamer,stkrause,kuefner,chvogt,zit}@tm.uka.de

21. Februar 2006

TM-2006-2

ISSN 1613-849X

http://doc.tm.uka.de/tr/

Institute of Telematics, University of Karlsruhe
Zirkel 2, D-76128 Karlsruhe, Germany

Vorwort

Das Seminar "Mobiles Internet" wurde im Wintersemester 2005/2006 in Form eines Blockseminars am 20. Februar 2005 am Institut für Telematik abgehalten. In diesem Seminarband sind die Ausarbeitungen der Studenten in Form eines internen Berichts zusammengefasst. Die behandelten Themen lassen sich in die folgenden vier Blöcke unterteilen.

IPv6 Movement Detection, Router Discovery und Adress-Konfiguration

Die IPv6-Protokolle für Movement Detection, Router Discovery und Adress-Konfiguration wurden im Hinblick den Einsatz in Netzen mit stationären Knoten entwickelt. Verzögerungen, die zum Zwecke von Ratenlimitierung und Desynchronization in die Protokolle eingebaut wurden, fallen daher sehr konservativ aus. Entsprechend ineffizient gestalten sich die Protokolle in mobilen Szenarien. Die Arbeiten zu diesem Thema analysieren die Leistungsfähigkeit der Standard-IPv6-Suite im Hinblick auf diese Verzögerungen und stellen den Vergleich zu Optimierungen an, die in der Forschungsgemeinschaft und IETF in jüngerer Zeit vorgeschlagen wurden.

Heterogene Mobilfunknetze der nächsten Generation

Es zeichnet sich immer stärker ab, dass die Mobilfunknetze der Zukunft durch Heterogenität geprägt sein werden. Die mobilen Endgeräte werden mit verschiedenen Funkschnittstellen, wie UMTS, WLAN und Bluetooth ausgerüstet sein, um in jeder Situation den günstigsten Zugang zum Internet zu nutzen. In diesem Seminar werden Erweiterungen von Mobile IPv6 vorgestellt, die es ermöglichen einzelne Datenströme zwischen verschiedenen Netzwerkschnittstellen zu bewegen (Flow Movement). Desweiteren wird der kommende IEEE-Standard 802.21 für medienunabhängige Handover genauer unter die Lupe genommen. Er soll insbesondere die Übergabe von Verbindungen zwischen verschiedenen Funktechnologien erleichtern. Schließlich wird die Entwicklung von UMTS hin zu einem All-IP-Netzwerk betrachtet. So bekommt der Leser einen Überblick darüber, wie die drei Standardisierungsgremien, IETF, IEEE und 3GPP mit der Heterogenität in zukünftigen Mobilfunknetzen umgehen.

Sicherheit in mobilen Ad-hoc-Netzen

Ein mobiles Ad-hoc-Netz (MANET) besteht aus mobilen Knoten, die über ein drahtloses Medium ein spontanes und dynamisches Netz aufbauen. Aufgrund der Mobilität der einzelnen Knoten ändert sich die Topologie des Netzes im Zeitablauf durch neue bzw. wegfallende Links. Um eine Kommunikation zwischen den teilnehmenden Knoten zu ermöglichen, wird ein MANET-Routingprotokoll eingesetzt. Derzeit existieren zwei Klassen von Routingprotokollen – reaktive und proaktive. Reaktive Protokolle, wie z.B. AODV, etablieren eine Route zwischen zwei Kommunikationspartner erst bei Bedarf. Proaktive Routingprotokolle, wie z.B. OLSR, sorgen im Gegensatz dazu dafür, dass alle Knoten eines Ad-hoc-Netzes eine komplette Sicht auf die Topologie besitzen und dadurch zu jedem möglichen Kommunikationspartner bereits im Voraus eine Route bekannt ist.

MANET Routingprotokolle wurden allerdings entwickelt, ohne sich Gedanken über Sicherheit zu machen, d.h. bei der Entwicklung wurde davon ausgegangen, dass alle Knoten sich gemäß den Protokollspezifikationen verhalten. Um ein Ad-hoc-Netz auch gegen bösartige Knoten zu sichern, müssen sowohl das Routingprotokoll als auch der Nutzdatenverkehr geschützt werden. Vor allem der sichere Austausch von Routing- und Topologie-Informationen wurde bisher

wenig beachtet und stellt daher ein häufig genutztes Angriffsziel dar. In diesem Seminar werden daher Sicherheitserweiterungen für die Routingprotokolle AODV und OLSR betrachtet. Ein weiteres Thema beschäftigt sich mit verteiltem Schlüsselmanagement in Ad-hoc-Netzen.

Mobilitätsunterstützung durch Overlays

Um im Internet mobile Knoten unterstützen zu können, sind Anpassungen oder Erweiterungen der bestehenden Architektur nötig. Neben der bekannten Möglichkeiten durch MobileIP gibt es die Alternative, Overlaynetze zu etablieren, die die benötigten Funktionaltäten breitstellen können. Dies bietet den Vorteil, dass neben einfacher Mobilität auch weitere neuartige Kommunikationsformen wie Multi- oder Anycast realisiert werden können, ohne dass die Kerninfrastruktur des Netzes geändert werden muss.

Ein Vertreter eines solchen Overlays ist das i3-Protokoll. Dieses wird in diesem Seminar in einer Arbeit vorgestellt, sowie dessen Stärken und Schwächen herausgearbeitet. Eine weitere Arbeit widmet sich Erweiterungen zu i3 und der Verbindung von i3 mit dem Host Identity Protokoll HIP, die ein Maximum an Flexibilität mit der größtmöglichen Sicherheit verbinden soll.

Inhaltsverzeichnis

Movement Detection und Router Discovery in IPv6 und Optimierungen

Jochen Gamer

Kurzfassung

Mobilitätsunterstützung in IPv6 erfordert die Erreichbarkeit eines mobilen Gerätes über eine eindeutige IP-Adresse und die Aufrechterhaltung bestehender Verbindungen beim Wechsel des Netzes. Hierbei ist es notwendig, mit Hilfe von Movement Detection solche Netzwechsel zuverlässig zu erkennen und im neuen Netz möglichst schnell den Aufbau von Verbindungen zu ermöglichen. Normale Router Discovery Mechanismen beinhalten hierbei verschiedene Verzögerungen, die gerade für Echtzeitanwendungen inakzeptabel sind. Diese Arbeit möchte die grundlegenden Mechanismen, ihre Nachteile und Verbesserungen vorstellen.

1 Einleitung

Diese Arbeit entstand im Rahmen des Seminars "Mobiles Internet" im Wintersemester 2005/06 am Lehrstuhl für Telematik an der Universität Karlsruhe (TH). Sie beschäftigt sich mit dem Zusammenhang zwischen der Erkennung von Bewegung eines mobilen Gerätes sowie dem Auffinden von Routern im Internet Protokoll IPv6. Zusätzlich werden Optimierungen der Standardvorgehensweisen vorgestellt.

Beim ersten Punkt, Movement Detection (2), wird zunächst darauf eingegangen, was Bewegung für ein Gerät bedeutet, welche Indizien für die Bewegung eines Gerätes gefunden werden können, wie zuverlässig diese Indizien tatsächlich sind und wie letztendlich sichergestellt werden kann, dass Bewegung korrekt erkannt wurde.

Router Discovery (3) beschäftigt sich damit, wie ein mobiles Gerät in dem Netzabschnitt, in dem es sich gerade befindet einen Router auffindet, wie es dieses Auffinden aktiv unterstützen kann und wie eine Verbindung zwischen Gerät und Router zustande kommt.

Der letzte Abschnitt (4) widmet sich den Schwachstellen der bisher vorgestellten Vorgehensweisen und stellt Optimierungen der Standardprotokolle wie FastRD und FastRA vor.

Zunächst jedoch möchte ich einige wiederkehrende Begriffe bzw. Abkürzungen nennen und einen kurzen Überblick geben, wie Mobilität in IPv6 unterstützt wird.

1.1 Begriffsdefinitionen

Mobile Node (MN): ein mobiles Gerät

Heimatadresse / Home Address: IP-Adresse des MN innerhalb des Heimatnetzes.

Home Agent: Router aus dem Heimnetz des MN, der für die Kommunikation des MN verantwortlich ist.

Foreign Agent: Router aus einem Fremdnetz, über den der MN kommuniziert.

Care-of-Address (CoA): IP-Adresse aus dem Fremdnetz, die dem MN durch übliche Mechanismen wie stateless oder stateful auto-configuration zugeordnet wurde.

Correspondent Node (CN): beliebiges Gerät, das mit dem MN kommuniziert.

1.2 Mobilität in IPv6

Von einem MN wird stets erwartet, über eine eindeutige IP-Adresse, seine Heimatadresse erreichbar zu sein, selbst wenn er sich nicht innerhalb des Heimatnetzes befindet. Bestehende Verbindungen sollen auch beim Wechsel eines Zugangspunktes aufrechterhalten werden (vgl. [JoPA04])

Solange ein MN sich im Heimatnetz befindet werden Pakete an die Heimatadresse des Gerätes über gewöhnliche Routing Mechanismen an das Gerät weitergeleitet. Befindet sich ein MN in einem fremden Netz, sucht er sich in diesem Fremdnetz einen Router (Abschnitt 3), der für ihn als “Standardrouter“ arbeitet (also seinen Foreign Agent). Innerhalb dieses Netzes ist der MN über eine oder mehrere CoAs erreichbar. Solange der MN in diesem Netz bleibt, werden an diese CoA adressierte Pakete an den MN weitergeleitet. Diese Verbindung zwischen CoA und Heimatadresse wird als “Binding“ bezeichnet und vom MN seinem Home Agent durch Senden eines “Binding Updates“ mitgeteilt und von diesem mit einem “Binding Acknowledgement“ bestätigt.

Für die Kommunikation zwischen MN und CN gibt es nun zwei Möglichkeiten (Vgl. Abbildung 1):

- “bidirectional tunneling“: Pakete vom CN an den MN werden zunächst an das Heimatnetz geschickt und dort vom Home Agent abgefangen und zum MN getunnelt. Pakete an den CN werden vom MN an den Home Agent getunnelt und dann normal vom Heimatnetz an den CN weitergeschickt.

- “Route optimization“ Hierfür ist es nötig, dass der MN sein aktuelles Binding dem CN mitteilt. Anschließend können Pakete vom CN direkt an die CoA des MN verschickt werden. Hierdurch wird ermöglicht, dass der beste Übertragungsweg zwischen CN und MN genutzt wird. Zudem werden die Auswirkungen möglicher Fehler beim Home Agent oder der Netzwerke auf dem Weg zu oder vom Heimatnetz des MN verringert. Pakete an den CN werden vom MN direkt verschickt, als Source Address wird nun die aktuelle CoA eingetragen.

Wechselt ein MN nun von einem Netz in ein anderes, so soll er trotz dieses Wechsels für bestehende Kommunikationspartner erreichbar bleiben. Hierfür gibt er den Netzwechsel seinem Home Agent über ein erneutes Binding Update bekannt. Beim bidirectional tunneling passt nun der Home Agent die Zieladresse für das Tunneln an, bei Route optimization ist es nötig, dass der MN dem CN die neue CoA (oder falls er in sein Heimatnetz zurückgekehrt ist den Wegfall einer solchen) mitteilt. Der Übergang von einem Netz in ein anderes und der damit verbundene Aktualisierungsaufwand werden als “Handover“ bezeichnet.

Im Folgenden geht es also zusammenfassend darum, zu beschreiben, wie ein MN üblicherweise bemerkt, dass ein Handover nötig ist (Abschnitt 2), wie er sein neues Netz erkennt und sich darin integriert (Abschnitt 3) und wie diese Prozesse beschleunigt werden können (Abschnitt 4).

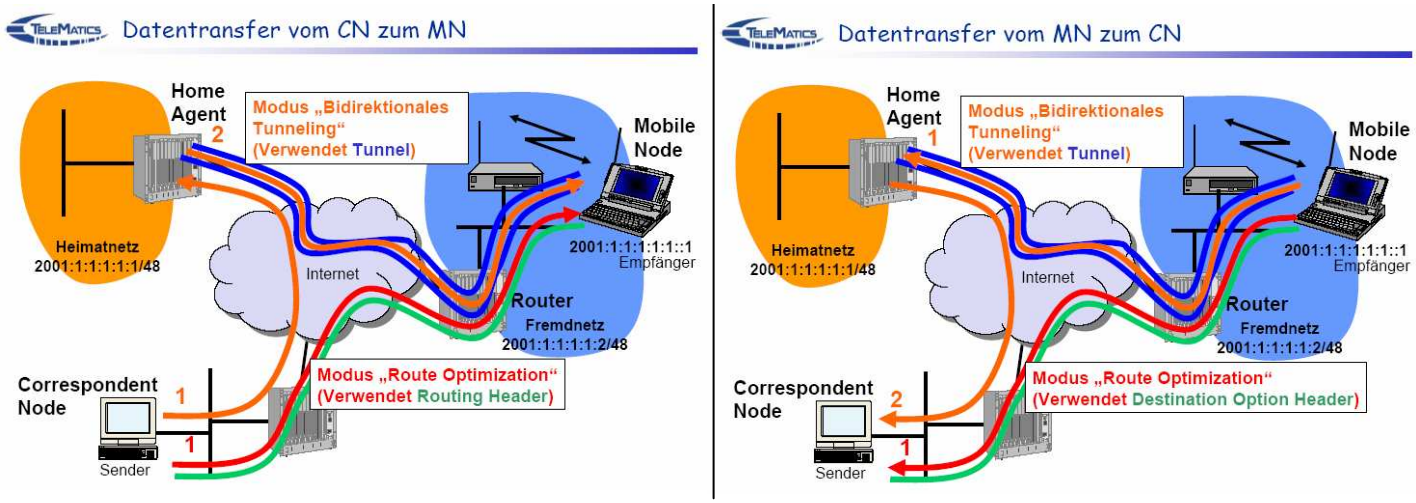

Abbildung 1: Kommunikation MN und CN, entnommen aus [Zitt05]

2 Movement Detection

Movement Detection beschäftigt sich damit, Handover auf Schicht 3 (L3-Handover) zu erkennen. Hierfür gilt es zunächst zu erarbeiten, auf welche Weise ein MN bemerkt, dass eine Bewegung vorliegen kann und wie zuverlässig diese Indizien für die Erkennung von L3-Handovern sind. Anschließend wird ein generischer Movement Detection Mechanismus zur gezielten Erkennung solcher L3-Handover vorgestellt.

2.1 Überblick

Um die kurzzeitige Unterbrechung des Paketflusses aufgrund eines L3-Handovers so gering wie möglich zu halten, sollten Bewegungen möglichst frühzeitig erkannt werden. Es reicht also nicht aus, wenn ein MN erst beim dauerhaften Wegfall der Verbindung zum aktuell genutzten Router Mechanismen zum Aufbau einer neuen Verbindung einleitet. Andererseits bedeutet jeder L3-Handover nicht nur eine kurzzeitige Unterbrechung der Verbindung sondern auch einen Signalisierungsaufwand z.B. durch Binding Updates. Es kann also auch nicht das Ziel sein, bei jedem Anzeichen für eine Bewegung sofort entsprechende L3-Handover durchzuführen.

Um unnötige L3-Handover zu vermeiden, aber nötige nicht zu verzögern, ist es also wichtig, entsprechende Anzeichen für die Bewegung eines MN in geeigneter Weise zu erkennen und zu behandeln

Üblicherweise gibt es drei Dinge, die sich bei einer Bewegung eines MN ändern und die somit als Indizien für die Notwendigkeit eines L3-Handovers in Frage kommen können (vgl. hierzu [Dale03], Kapitel 2.2)

1. Informationen niedrigerer Schichten über Verbindungsänderungen
2. Das Auffinden eines Routers mit neuer IP-Adresse
3. Das Auffinden eines Routers mit neuem Subnetz Präfix

(Details über das Auffinden von neuen Routern finden sich in Abschnitt 3)

Um den Zusammenhang zwischen dem Auftreten dieser möglichen Indizien und der Erfordernis eines L3-Handovers zu veranschaulichen, möchte ich im Folgenden einige Szenarien darstellen:

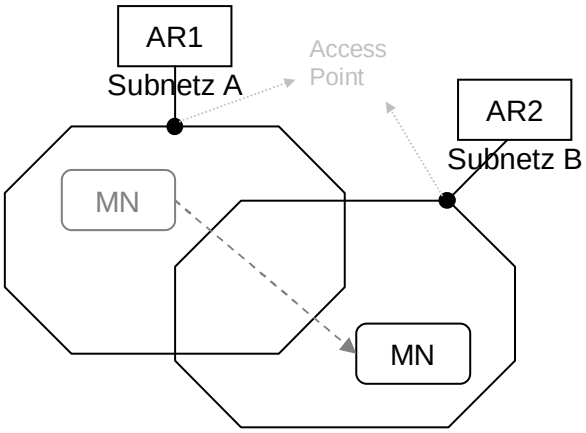

Abbildung 2: MN verlässt altes Netz und wechselt in neues Subnetz

In Abbildung 2 bewegt sich der MN von einem Subnetz in ein anderes. Hier liegen sowohl Informationen niedriger Schichten über Verbindungsänderungen (nämlich der Wechsel des Access Points auf Schicht 2) als auch ein neuer Access Router (AR) mit geänderter IP-Adresse und anderem Netzpräfix vor. In diesem Fall liegen also alle drei oben genannten Indizien vor und ein L3-Handover wird benötigt.

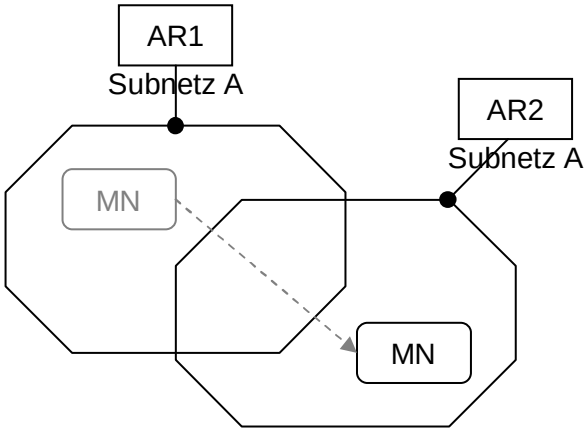

Abbildung 3: AR1 und AR2 sind Router im selben Netz

Bei Abbildung 3 sind AR1 und AR2 Router im selben Netz. Wechselt der MN nun von AR1 zu AR2, so wechselt er Access Point und Access Router, bleibt jedoch im selben Subnetz.

Im Falle von Abbildung 4 ist ein neuer AR mit neuer IP-Adresse und neuem Netzpräfix über einen neuen Access Point erreichbar. Liegt auf Schicht 2 jedoch noch kein Wechsel des Access Points vor, so ist dieser AR für den MN noch nicht sichtbar. Im vorliegenden Fall entscheidet also ein L2-Handover über die Notwendigkeit eines L3-Handover. Dies sollte allerdings vermieden werden, da das alte Netz und damit der bisherige AR noch erreichbar sind.

In Abbildung 5 teilen sich AR1 und AR2 einen Access Point. Ist AR1 der aktuelle default Router des MN bedeutet das Auftauchen von AR2 keine Bewegung des MN.

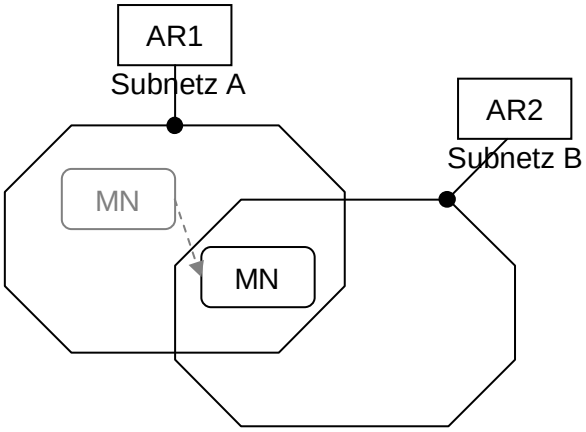

Abbildung 4: MN kommt in neues Subnetz, altes Netz bleibt jedoch erreichbar

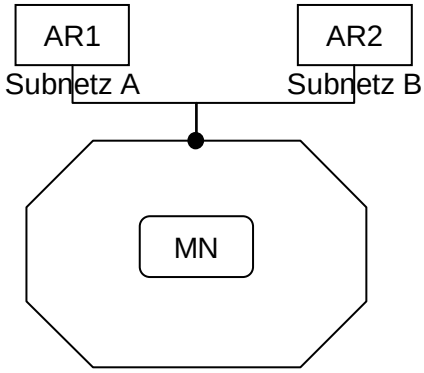

Abbildung 5: Zwei Router im gleichen Subnetz

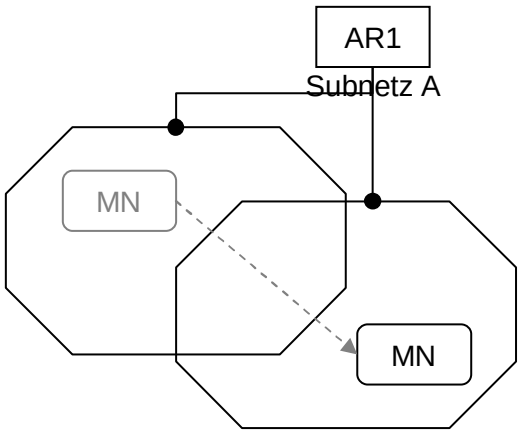

Abbildung 6: MN wechselt den Zugangspunkt innerhalb eines Netzes

Bei Abbildung 6 wechselt der MN von einem Access Point zu einem anderen. Dennoch bleibt er trotz der Meldung einer Bewegung von Schicht 2 im selben IP-Subnetz.

Ein Movement Detection Mechanismus sollte alle verfügbaren Informationen zur korrekten Erkennung von Bewegung hinzuziehen. Dennoch sollte nicht aufgrund irgendwelcher Hinweise Bewegung erkannt werden, wenn der MN sich nicht bewegt hat. Folgende Punkte sollten daher beachtet werden:

Über eine Verbindung können mehrere Router erreichbar sein, das Auffinden eines neuen Routers erfordert nicht zwangsläufig einen L3-Handover Senden verschiedene Router über eine Verbindung, so können diese ein anderes Netzpräfix verwenden. Selbst das Auffinden eines neuen Routers mit neuem Netzpräfix ist daher kein zuverlässiges Anzeichen für Bewegung. Ein Router kann über mehrere Access Points verfügbar sein. Ein L2-Handover lässt zwar darauf schließen, dass der MN sich bewegt, es muss jedoch nicht zwangsläufig ein L3-Handover nötig sein. Die Adressen von Routern sind verbindungsspezifische Adressen und daher nicht global eindeutig. Selbst wenn der MN Pakete von derselben Adresse empfängt, ist nicht sichergestellt, dass sie auch vom bisherigen Access Router stammen (Dieses Problem kann allerdings vermieden werden, wenn Router das Router Address Bit nutzen, da dies eine global eindeutige Adresse bietet).

2.2 Movement Detection Mechanismus

Ein Movement Detection Mechanismus sollte die oben genannten Hinweise, aber auch die dabei auftretenden Schwierigkeiten bei der zuverlässigen Erkennung benötigter L3-Handover berücksichtigen. Ein generischer Ansatz für einen Movement Detection Mechanismus (in Anlehnung an [Dale03], Kapitel 2.3) ist es, ausgehend von vorhandenen Hinweisen zunächst die tatsächliche Erreichbarkeit des aktuellen Access Routers zu testen und anschließend gegebenenfalls die Gültigkeit der CoA zu überprüfen. Wird festgestellt, dass der bisherige Access Router nicht erreichbar und/oder die CoA nicht mehr gültig ist, so muss ein neuer Access Router gefunden werden.

2.2.1 Auffinden von Hinweisen auf Bewegung

Zusätzlich zu den bereits im Überblick genannten Hinweisen ist auch das Fehlen eines regelmäßigen Signals bzw. ein Ablauf der Gültigkeit des aktuellen Access Routers zu berücksichtigen. In diesen Fällen ist die Erreichbarkeit des aktuellen Access Routers mit Hilfe von Neighbor Discovery zu testen.

2.2.2 Testen der Erreichbarkeit des aktuellen Access Routers

Ein MN testet die Erreichbarkeit des aktuellen Routers mit Hilfe von Neighbor Solicitation (NS) Nachrichten. Hierbei schickt er eine solche Nachricht an den aktuellen Router, erhält er ein Neighbor Acknoledgement (NA), kann er davon ausgehen, dass der Router unter dieser Adresse immer noch erreichbar ist. Erhält der MN innerhalb einer gewissen Zeitspanne kein NA, so ist der Router schlecht oder nicht erreichbar. In diesem Fall können entweder weitere NS verschickt werden oder der MN kann direkt davon ausgehen, dass er sich bewegt hat.

2.2.3 Überprüfen der Gültigkeit der aktuellen CoA

Hat ein MN den Access Router gewechselt, so ist die bisherige CoA ungültig. Doch auch wenn der MN beim Testen der Erreichbarkeit des aktuellen Access Routers (Kapitel 2.2.2) ein gültiges NA erhalten hat, kann er nicht immer davon ausgehen, dass diese Antwort auch tatsächlich vom zuvor genutzten Router stammt. Da die Eindeutigkeit von Link-Local-Adressen nur innerhalb eines Subnetzes garantiert ist, ist es möglich, dass der MN sich in ein anderes IP-Subnetz zu einem neuen Router, der dieselbe lokale Adresse besitzt, bewegt hat. Daher muss in diesem Falle die Gültigkeit der aktuellen CoA überprüft werden. Hierzu sendet der MN eine Router Solicitation (RS) Nachricht zum neuen Router, auf die dieser mit einem Router Advertisement (RA) antwortet (Details hierzu in Kapitel 3). Diesem RA kann der MN

nun das Subnetzpräfix entnehmen und es mit dem bisherigen vergleichen. Weicht es ab, so hat sich der MN in ein anderes Subnetz bewegt und die bisherige CoA ist nicht mehr gültig.

2.2.4 Auffinden eines neuen Access Routers

Wurde beim Testen der Erreichbarkeit des Access Routers und beim Überprüfen der Gültigkeit der CoA festgestellt, dass zwar ein Router unter der bisherigen Adresse gefunden wird, dieser aber zu einem anderen Subnetz gehört, so wird dieser Router als neuer Access Router genutzt und die CoA über Binding Updates aktualisiert. Wurde bei 2.2.2 kein Router - also weder der alte noch ein neuer unter dieser Adresse- gefunden, so ist es notwendig, dass der MN möglichst schnell einen neuen Access Router findet, um die Unterbrechung der aktuellen Verbindungen möglichst gering zu halten. Wie ein Gerät verfügbare Router erkennt und wie es danach bei Bedarf aktiv suchen kann ist Thema des nächsten Kapitels.

3 Router Discovery

Das Auffinden von verfügbaren Routern kann sowohl vom Router als auch vom Host ausgehen. Geräte, die als Router fungieren versenden regelmäßige Unsolicited Router Advertisements (RAs) durch Multicasting. Anhand dieser Pakete erkennen Hosts innerhalb weniger Minuten, welche Router für sie erreichbar sind. Sie reichen jedoch nicht aus, einen Ausfall von Routern zuverlässig zu erkennen. RAs enthalten eine Liste von Subnetzpräfixen sowie Präfix-Flags, die zur Adresskonfiguration genutzt werden können. Zusätzlich umfassen sie Internet Parameter wie z.B. Hop Limit und optionale Linkparameter

Benötigt ein Host innerhalb kurzer Zeit einen verfügbaren Router, z.B. wenn er neu angeschlossen wurde, nach einem Ausfall neu startet oder aufgrund von Bewegung Router in einem neuen Subnetz finden muss, so kann er über eine Router Solicitation Nachricht erreichbare Router dazu auffordern, ihre Verfügbarkeit über außerplanmäßige Router Advertisements anzuzeigen.

Für beide Arten zur Erkennung verfügbarer Router werden nun die Abläufe auf Seiten des Routers und des Hosts sowie die Nachrichtenformate vorgestellt.

3.1 Unsolicited Router Advertisements

Unsolicited RAs dürfen nur von Routern gesendet werden. Sie werden regelmäßig, jedoch nicht nach immer gleichen Zeitabständen versendet. Das Intervall zwischen einzelnen Übertragungen wird zufällig bestimmt um die Synchronisation mit Advertisements anderer Router zu vermeiden. Nach jedem gesendeten RA wird der Timer bis zum nächsten Versenden auf einen zufälligen Wert innerhalb der im Router definierten Variablen MinRtrAdvInterval und MaxRtrAdvInterval zurückgesetzt.

Die ersten Ras (bis zu MAX_INITIAL_RTR_ADVERTISEMENTS) die ein Gerät sendet, wenn es zum Router wird (z.B. durch entsprechende Konfiguration oder Neustart) sollten in relativ kurzen Zeitabständen erfolgen, um die Wahrscheinlichkeit zu erhöhen, von Hosts entdeckt zu werden. Hierfür ist ein Wert MAX_INITIAL_RTR_ADVERT_INTERVAL festgelegt, der den maximalen Zeitabstand zwischen zwei RAs in der Initialisierungsphase des Routers bestimmt. Diese Regelungen gelten auch, wenn die Informationen innerhalb des RA wie z.B. die Lebenszeit der Präfixe sich ändern, neue Präfixe hinzukommen oder das Gerät die Routing Funktionen einstellt (siehe hierzu auch [NNSS05], Kapitel 6.2.4).

3.1.1 Umgang mit Unsolicited Router Advertisements

Erhält ein Host ein gültiges RA, so entnimmt er diesem die Source Address und handelt folgendermaßen:

- Enthält die Default Router List des Hosts die Adresse nicht und die Lebenszeit des Routers im RA ist nicht Null, so wird ein neuer Eintrag in der Liste erstellt und die Ablaufzeit entsprechend des Lifetime-Feldes des RA gesetzt.
- Enthält die Default Router List des Hosts die Adresse bereits, so wird die Ablaufzeit entsprechend des Lifetime-Feldes des RA aktualisiert.
- Enthält die Default Router List des Hosts die Adresse bereits und die Lebenszeit des Routers ist Null, so wird der Router aus der Liste entfernt.

Ein Host muss nicht zwingend jeden neuen Router in die Default Router List aufnehmen, es müssen jedoch mindestens zwei Router Adressen geführt werden und es sollten mehr sein. Default Router werden immer dann ausgewählt, wenn bei der Kommunikation Probleme auftreten. Je mehr Router in der Liste enthalten sind, desto größer ist in diesem Fall die Wahrscheinlichkeit schnell eine funktionierende Alternative beim Ausfall eines oder mehrerer Router zu finden ohne auf die Ankunft des nächsten RA warten zu müssen. Eine lange Default Router List hilft jedoch nicht beim Auffinden eines neuen Routers nach einem IP-Movement, da in diesem Fall alle Einträge der Liste ungültig werden (siehe hierzu auch [NNSS05], Kapitel 6.3.4).

3.1.2 Unsolicited Router Advertisements Nachrichten Format

Unsolicited Router Advertisements Nachrichten sind wie in Abbildung 7dargestellt aufgebaut.

0										1										2										3	
0 1		2 3		4 5		6 7		8 9		0 1		2 3		4 5		6 7		8 9		0 1		2 3		4 5		6 7		8 9		0 1	
Type								Code								Checksum															
Cur Hop Limit								M | O		Reserved									Router Lifetime											
Reachable Timer																															
Retrans Timer																															
Options...																															

Abbildung 7: Unsolicited Router Advertisements Nachrichten Format

M: 1-bit “Managed address configuration“ flag. Zeigt an, dass das Dynamic Host Configuration Protocol [DHCPv6] zur Adress-Konfiguration zur Verfügung steht.

O: 1-bit “Other configuration“ flag. Zeigt an, dass DHCPv6lite zur Konfiguration anderer Informationen zur Verfügung steht.

Detailliertere Informationen zur Bedeutung der einzelnen Felder finden sich in [NNSS05], Kapitel 4.2).

3.2 Router Solicitations

Benötigt ein Host möglichst schnell einen verfügbaren Router und möchte nicht bis zum nächsten Unsolicited RA warten, so kann er bis zu MAX_RTR_SOLICITATIONS Router Solicitation (RS) Nachrichten versenden. Diese müssen mindestens RTR_SOLICITATION_INTERVAL Sekunden auseinander liegen und können in einem der folgenden Fälle verschickt werden:

- Das Interface wurde beim Systemstart initialisiert.
- Das Interface wurde nach einem Fehler oder einer vorübergehenden Deaktivierung neu gestartet.
- Das System wird vom Router zum Host.
- Der Host ist zum ersten Mal an einen Link angebunden.
- Der Host wird an einen Link erneut angebunden, nachdem er einige Zeit getrennt war.

RS werden vom Host an die Router Multicast Adresse gesendet. Die IP Quelladresse ist -sofern vorhanden- eine der Unicast-Adressen des Interfaces oder eine unbestimmte Adresse. Wurde eine Adresse angegeben, so sollte auch die link-layer Quelladresse mitgeteilt werden.

Bevor ein Host eine erste RS verschickt, sollte er eine zufällige Zeitspanne zwischen 0 und MAX_RTR_SOLICITATION_DELAY abwarten. Dies verhindert einen Stau, wenn viele Hosts zur gleichen Zeit an einem Link in Betrieb gehen, beispielsweise nach einem Stromausfall. Wurde beim Host bereits aufgrund anderer Protokolle eine Verzögerung ausgelöst, so ist keine weitere mehr nötig. Werden viele RS innerhalb einer kurzen Zeitspanne verschickt, so kann dies (auch aufgrund der antwortenden RAs) ein Netz stark belasten und zu Engpässen führen. Unnötige oder aufgrund schwacher Bewegungshinweise versandte RS sollten daher vermieden werden.

Erhält ein Host nach Versenden eines RS ein gültiges RA dessen Router Lifetime nicht Null ist, so darf er keine weiteren RS verschicken bis einer der oben genannten Fälle eintritt. Darüber hinaus sollte ein Host mindestens ein RS verschicken, selbst wenn er bereits ein RA empfangen hat, da Solicited RAs alle erforderlichen Informationen enthalten, die teilweise in Unsolicited RAs fehlen dürfen.

Bekommt ein Host auch nach MAX_RTR_SOLICITATIONS Nachrichten kein RA, so geht er nach MAX_RTR_SOLICITATION_DELAY Sekunden davon aus, dass auf dem aktuellen Link keine Router erreichbar sind. Dennoch bleibt er bereit zum Empfang und zur Bearbeitung ankommender RA für den Fall, dass Router auf diesem Link auftauchen.

Siehe hierzu auch [NNSS05], Kapitel 6.3.7.

3.2.1 Umgang mit Router Solicitations

Hosts müssen empfangene RS verwerfen.

Router senden RAs zusätzlich zu regelmäßigen Unsolicited RAs (Kap 3.1) auch als Antwort auf erhaltende gültige RS. Hierbei kann ein Router wählen, ob er diese Antwort als Unicast direkt an den suchenden Host schickt (sofern eine gültige Quelladresse des RS vorliegt), normalerweise jedoch sendet er ein RA über Multicast an alle Geräte. In diesem Fall wird der Intervall Timer bis zum Versenden des nächsten Unsolicited RA auf einen neuen Zufallswert zurückgesetzt.

Alle Antworten auf erhaltene RS müssen für eine bestimmte Zeit zwischen 0 und MAX_RA_DELAY_TIME Sekunden verzögert werden um auch hier Überschneidungen von RA mehrerer Router zu vermindern. Zudem darf nicht mehr als ein RA per Multicast innerhalb MIN_DELAY_BETWEEN_RAS Sekunden verschickt werden.

Router Solicitations können vom Router folgendermaßen behandelt werden:

- Beim Empfang eines RS wird ein Zufallswert zwischen 0 und MAX_RA_DELAY_TIME bestimmt. Ist vor Ablauf dieses Timers bereits der Versand eines regulären Multicast RA vorgesehen, so wird dies wie geplant verschickt.
- Hat der Router bereits innerhalb der letzten MIN_DELAY_BETWEEN_RAS ein Multicast RA verschickt, so wird das neue RA MIN_DELAY_BETWEEN_RAS plus dem Zufallswert nach dem letzten RA versandt, um sicherzustellen, dass die Einschränkungen bei der Übertragung eingehalten werden.
- In allen anderen Fällen wird der Versand nach Ablauf des Zufallswertes eingeplant.

Siehe hierzu auch [NNSS05], Kapitel 6.2.6.

3.2.2 Router Solicitations Nachrichten Format

Router Solicitations Nachrichten sind wie in Abbildung 8dargestellt aufgebaut.

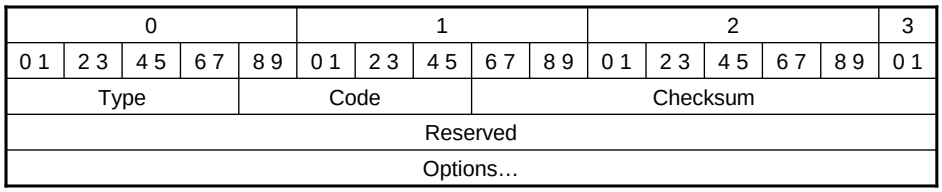

Abbildung 8: Router Solicitations Nachrichten Format

Detailliertere Informationen zur Bedeutung der einzelnen Felder finden sich in [NNSS05], Kapitel 4.1).

4 Optimierungen

Wie in den vorherigen Kapiteln dargestellt, gibt es beim Übergang eines MN von einem Netz in ein anderes zahlreiche Verzögerungen. Zum einen erzwingt das Neighbor Discovery Protokoll von Routern eine minimale Wartezeit von 3 Sekunden zwischen dem Senden von Multicast RA Nachrichten. Des Weiteren soll ein Host bei der Initialisierung den Versand von RS eine zufällige Zeitspanne verzögern. Ebenso muss ein Router mit der Antwort auf ein RS eine zufällige Zeitspanne warten. Die daraus resultierende Wartezeit beim L3-Handover ist insbesondere für Echtzeitanwendungen wie z.B. VoIP inakzeptabel. Im Folgenden sollen zwei Ansätze vorgestellt werden, mit denen diese Wartezeit verringert werden kann. Fast Router Advertisements (FastRA) stellt eine Möglichkeit dar, wie die Bearbeitungszeit von RS bei Routern deutlich verkürzt werden kann, Fast Router Discovery (FastRD) ist ein Ansatz, bei dem mit Hilfe von Schicht 2-Funktionen der Zugang von RA Nachrichten beschleunigt wird.

4.1 Fast Router Advertisements

Um schnellere Antwortzeiten bei der Bearbeitung von RS Nachrichten zu ermöglichen, sollte zumindest ein Router ohne Wartezeit auf ein RS eines Hosts antworten dürfen. Ein RA, das direkt ohne Verzögerung durch Unicast an den Host verschickt wird, wird als Fast RA bezeichnet (vgl. [KeKP04]).

Ein Router, der zur Erstellung von FastRAs konfiguriert ist, muss über einen Zähler FastRACounter die Anzahl der gesendeten FastRAs seit dem letzten regulären Unsolicited RA bestimmen. Erhält der Router ein RS, so muss er sofort ein RA verschicken, sofern der Zähler geringer ist als MAX_FAST_RAS (die maximale Anzahl zulässiger FastRAs zwischen Unsolicited RAs). Sofern die Quelladresse des RS bekannt ist, sollte der Router die Antwort direkt über Unicast an den Host schicken, andernfalls muss ein reguläres Multicast RA verschickt werden, bei dem die üblichen zeitlichen Einschränkungen eingehalten werden müssen (vgl. Kapitel 3.2.1). Immer wenn ein FastRA verschickt wird, wird der Zähler FastRACounter um eins erhöht. Wird ein Unsolicited RA verschickt, so wird der Zähler wieder zurückgesetzt. Erreicht der FastRACounter den maximal zulässigen Wert MAX_FAST_RAS, so muss der Router schnellstmöglich ein Multicast RA einplanen (auch hier müssen die oben genannten Einschränkungen eingehalten werden). Bis zum Versand dieses Multicast RA dürfen keine weiteren RS beantwortet werden.

Vorteil dieses Ansatzes ist die schnelle Antwortzeit durch einen ausgewählten Router. Weitere Router zum Aufbau einer Default Router List werden mit etwas Verzögerung gefunden. Problematisch ist jedoch die Festlegung, welcher Router an welchen Host FastRA Nachrichten verschicken darf. FastRA bietet nämlich nur dann eine wirkliche Verbesserung, wenn sichergestellt werden kann, dass für jeden Host bei jedem Zugangspunkt ein Router diese Möglichkeit anbietet. Überschneiden sich zwei oder mehr FastRAs unterschiedlicher Router, so laufen diese synchron und können sich gegenseitig behindern.

4.1.1 DNAv6

Das oben genannte Problem lässt sich mit Hilfe von DNAv6 (Detecting Network Attachment in IPv6 Networks, siehe hierzu [Pent05]) verbessern. Hierbei sammelt jeder Router die link-local Adressen aller anderen Router im selben Netz. Sendet nun ein Host ein RS, so berechnet jeder Router daraus ein Host Token. Anhand dessen kann nun über eine XOR-Operation mit den Router Token für jeden Router ein Wert berechnet werden. Diese Werte werden sortiert und bestimmen die Reihenfolge, in der die Router RAs an den Host schicken. Der Router an Position 0 schickt ohne Verzögerung, jeder andere multipliziert seine Position mit einer festgelegten Verzögerungszeit “RASeparation“ und wartet entsprechend lange bis zum Versand seines RAs. Hierbei ist es nicht nötig, dass jeder Router die komplette Liste sortiert. Es reicht aus, wenn jeder Router seine eigene Position bestimmt.

Es ist auch hier nicht auszuschließen, dass mehrere Router aufgrund von Fehlern in der Router-Liste gleichzeitig RAs versenden, doch sind diese Fehler mit der Zeit sehr gering. Zudem stellt dieser Mechanismus sicher, dass nicht immer der gleiche Router als erstes auf ein RS antwortet und somit von vielen Hosts als bevorzugten Default Router genutzt wird, da für jeden Host eine andere Sortierung entsteht.

4.2 Fast Router Discovery

Bei Fast Router Discovery (vgl. [ChSh05]) wird eine Beschleunigung des Zugangs von RA Nachrichten Über PoAs (Point of Attachment) erreicht. Dies sind Zugangspunkte wie z.B. Access Points gemäß 802.11. Hierbei wird versucht, unmittelbar nach Aufbau einer L2 Verbindung ein Unicast RA an den Host zu verschicken. Hierfür kann der PoA entweder einen Access Router auffordern, ein gültiges RA sofort zu verschicken (“RA Triggering“) oder er kann ein solches selbst verschicken (“RA Proxying“).

Dieser Ansatz bietet für jede aufgebaute Verbindung auf Schicht 2 innerhalb kürzester Zeit die benötigten Informationen für den Aufbau eines Zugangs auf Schicht 3. Dies kann zwar dazu

führen, dass auch dann ein L3-Handover stattfindet, wenn dieser nicht zwingend notwendig ist (nämlich dann, wenn der alte Access Router auch über die neue Schicht 2 Verbindung erreichbar ist), doch ist dies aufgrund der sehr schnell verfügbaren Informationen und der geringen Netzbelastung verschmerzbar. Als problematisch ist jedoch die Verschmelzung von Schicht 2 und Schicht 3 Funktionen anzusehen.

4.2.1 RA Triggering

Befinden sich PoA und Access Router im selben Gerät, so kann der PoA bei der Anbindung eines neuen Host diese Verbindungsinformationen von Schicht 2 an Schicht 3 weitergeben. Mit diesen Informationen kann der Access Router ein gültiges RA erstellen und es sofort über eine Unicast Nachricht auf Schicht 2 an die MAC-Adresse des Host versenden. Sind PoA und Router getrennt, so können die nötigen Informationen vom PoA an den Access Router geschickt werden (beispielsweise über den Media Independent Event Service, MIES). Deser kann nach Erhalt der Informationen nun ebenfalls ohne Verzögerung ein passendes RA an den Host verschicken.

4.2.2 RA Proxying

RA Proxying beinhaltet RA Caching und RA Delivery. RA Caching dient dazu, ein gültiges RA zu erhalten und zu speichern. RA Delivery soll den sofortigen Unicast Versand eines gecachten RA an einen neuen Host sicherstellen.

RA Caching kann sowohl manuell als auch mit Hilfe von Scans durchgeführt werden. Access Router verschicken regelmäßig ein gültiges RA, das auch über den PoA übertragen wird. Daher besteht für den PoA die Möglichkeit, von eingehenden Nachrichten Level2-Informationen auszuwerten und geeignete RA zwischenzuspeichern. Dies kann kontinuierlich oder in regelmäßigen Abständen erfolgen, um das gespeicherte RA aktuell zu halten. Zusätzlich kann ein Access Router einem PoA beispielsweise über MICS (Media Independent Command Service) die nötigen Informationen zukommen lassen, so dass der PoA in der Lage ist, selbst gültige RAs zu erzeugen.

Sobald auf Schicht 2 eine neue Verbindung zu einem Host aufgebaut wurde kann der PoA das RA aus seinem Cache an diesen Host senden.

Literatur

[ChSh05] JinHyeock Choi und DongYun Shin. Fast Router Discovery with L2 support. Internet-Draft, Juli 2005.

[Dale03] Greg Daley. Movement Detection Optimization in Mobile IPv6. Internet-Draft, Mai 2003.

[JoPA04] D. Johnson, C. Perkins und J. Arkko. Mobility Support in IPv6. RFC 3775, Juni 2004.

[KeKP04] J. Kempf, M. Khalil und B. Pentland. IPv6 Fast Router Advertisement. Internet-Draft, Juli 2004.

[NNSS05] T. Narten, E. Nordmark, W. Simpson und H. Soliman. Neighbor Discovery for IP version 6. Internet-Draft, Oktober 2005.

[Pent05] B. Pentland. Detecting Network Attachment in IPv6 Networks. Internet-Draft, Oktober 2005.

[Zitt05] M. Zitterbart. Mobilkommunikation. Vorlesung, 2005.

Abbildungsverzeichnis

Adress-Konfiguration in IPv6 und Optimierungen

Stefan Kostov

Kurzfassung

Bei IPv6 unterscheidet man zwischen *zustandslose* und *zustandsbehaftete* Autokonfiguration. Während man mit zustandsbehafteter Adress-Autokonfiguration in der Regel den Einsatz von DHCPv6 meint, was es bei IPv4 in ähnlicher Weise auch gibt, bedeutet zustandslose Adress-Autokonfiguration einen Autokonfigurations-Mechanismus, der bei IPv6 völlig neu ist, und für den es bei IPv4 kein Äquivalent gibt. Heute wird in der Regel entweder die so genannte *zustandslose Address-Autoconfiguration* eingesetzt, oder man verwendet keine Autokonfiguration und konfiguriert Adressen manuell. Während es sicher Argumente gibt, die in manchen Fällen gegen die zustandslose Autokonfiguration sprechen, ist sie doch die einfachste Art und Weise ein ganzes Netz mit IPv6 zu adressieren und zu betreiben. Im Folgenden wird näher auf diesen Mechanismus eingegangen.

1 Einführung

Um den Standard für DHCPv6 [DBVL+03] wurde lange Zeit diskutiert und er wurde im RFC 3315 erst im August 2003 verabschiedet. Der DHCPv6-Mechanismus ist wesentlich mächtiger als sein Vorgänger – das DHCP für IPv4 und ist neben der einfachen Zuweisung von IPv6-Adressen u.A. auch in der Lage temporäre Adressen zu vergeben oder Adressen von Nameservern und NTP-Servern zu verteilen.

Die zustandslose Adress-Autokonfiguration [ThNJ05] wird in RFC 2462 beschrieben. Zur Verwendung dieses Mechanismus werden weder spezielle Klienten noch Server benötigt, denn die zustandslose Adress-Autokonfiguration zum IPv6-Standard gehört. Jede Plattform, die IPv6 implementiert verfügt über die benötigten Funktionen und für eine zustandslose Adress-Autokonfiguration auf einem Host sind keinerlei vorhergehende Konfigurationen notwendig. Die Verwendung der *zustandslosen Adress-Autokonfiguration* ist vorgegeben.

1.1 Warum eigentlich Duplicate Address Detection?

IPv6 Knoten sind in der Lage zustandslos eine eigene IPv6-Adresse zu konfigurieren. Die Konfigurierung derselben IPv6-Adresse von zwei verschiedenen Knoten könnte dann erhebliche Folgen haben. Die IPv6-Neighbour-Discovery-Mechanismen bieten eine gewisse Robustheit und das Netzwerk wird nicht instabil, durch eine solche falsche Konfigurierung. Allerdings werden diese beiden Knoten für die Adresse „kämpfen“, falls die Kollision unentdeckt bleibt. Beide Knoten werden also auf Neighbour Solicitations antworten und die jeweiligen Kommunikationspartner müssen die Antworten beliebig wählen, abhängig davon welche Antwort zuerst ankommt. Weiterhin könnten Pakete an den falschen Knoten geliefert werden, was wiederum bedeutet, dass er darauf falsch antworten könnte. Der kollidierende Knoten könnte z.B. fälschlicherweise TCP-Reset oder ICMP-Destination Unreachable zurücksenden, was zur Unterbrechung der bestehenden Verbindung führen würde.

Das eigentliche Problem ist aber, dass keiner der beiden Knoten die konfigurierte Adresse automatisch aufgeben und dekonfigurieren wird. Das führt wiederum zu Schwierigkeiten zur Lokalisierung des Problems, insbesondere außerhalb des lokalen Netzes, wo sich die beiden Knoten befindet.

Um diese mögliche Konflikte zu vermeiden werden Duplicate Address Detection Mechanismen im IPv6-Internet benötigt. Zunächst wird das DAD Protokoll vorgestellt mit seinem Vor- und Nachteile. Im zweiten Kapitel werden dann die Erweiterungen von DAD vorgestellt und analysiert.

1.2 IPv6 Adress-Autokonfiguration

Eine IPv6-Adress-Autokonfiguration [ThNJ05] kann nur an Multicast-fähige Links durchgeführt werden und startet immer dann, wenn ein Multicast-fähiger Interface aktiviert wird, z.B. beim System-Hochfahren. Jedes Interface beginnt den Autokonfigurationsprozess mit der Konfigurierung einer so genannten Link-Local-Adresse, indem er aus seiner MAC-Adresse (EUI-48) eine EUI-64-ID erzeugt und diese an das Präfix für Link-Lokal-Adressen (fe80::/10) anhängt:

```
fe80::<erste Hälfte der MAC-Adresse>ff:fe<zweite Hälfte der MAC-Adresse>/10
```

Bevor aber die Link-Local-Adresse an dem Interface zugewiesen wird und benutzt werden darf ist sie als *tentative (provisorisch)* Bezeichnet. Der jeweilige Knoten muss noch überprüfen ob diese Adresse auf dem Link eindeutig ist. Der zugehörige Mechanismus nennt sich *Duplicate Address Detection* oder kurz *DAD* und funktioniert folgendermaßen:

Zunächst sendet der Knoten ein Neighbour Solicitation, das als Zieladresse die tentative Adresse enthält. Falls ein anderer Knoten diese Adresse schon verwendet, antwortet er mit einem Neighbour Advertisement. In diesem Fall bricht der komplette Autokonfigurationsprozess ab und ein manueller Eingriff ist nötig. Die Zeit, die vergehen muss, um sicherzustellen, dass die konfigurierte Link-Local-Adresse verwendet werden darf ist Link-spezifisch und wird durch die folgenden zwei Parameter bestimmt: der Knoten muss das Neighbour Solicitation insgesamt *DupAddrDetectTransmits* mal wiederholen und der Zeitabstand zwischen zwei Wiederholungen beträgt jeweils *RetransTimer* Millisekunden. Vorgegeben sind eine Wiederholung und 1000 Millisekunden Widerholungszeit (siehe Abbildung 1).

Falls kein Neighbour Advertisement zurückkommt, bedeutet dies, dass die Link-Lokal-Adresse auf dem Link einzigartig ist und sie darf fest auf das Interface konfiguriert werden. Ab diesem Zeitpunkt hat der Knoten die Möglichkeit mit anderen Knoten auf demselben Link über IPv6 zu kommunizieren.

Dieser erste Teil der Adress-Autokonfiguration wird immer auf allen IPv6-Netzwerkknoten ausgeführt, egal ob es sich um einen Host oder um einen Router handelt, oder ob im Folgenden DHCPv6 für die Konfiguration weiterer Adressen verwendet werden soll oder nicht. Außer der eigenen MAC-Adresse wird dazu nichts weiter benötigt, nicht einmal die Verbindung zu einem echten LAN. Die folgenden Schritte werden nur auf Hosts durchgeführt. Die Konfiguration von Routern erfordert ab dieser Stelle manuelles Eingreifen oder die Benutzung spezieller Funktionalitäten von DHCPv6, die in diesem Dokument nicht weiter beschrieben werden sollen.

Nach der Konfiguration der Link-Lokal-Adressen, muss ein Host zunächst feststellen ob auf dem gleichen Link Router existieren, die Router-Advertisements versenden. Je nach Router-Konfiguration werden die Router-Advertisements in bestimmten Zeitabständen verschickt.

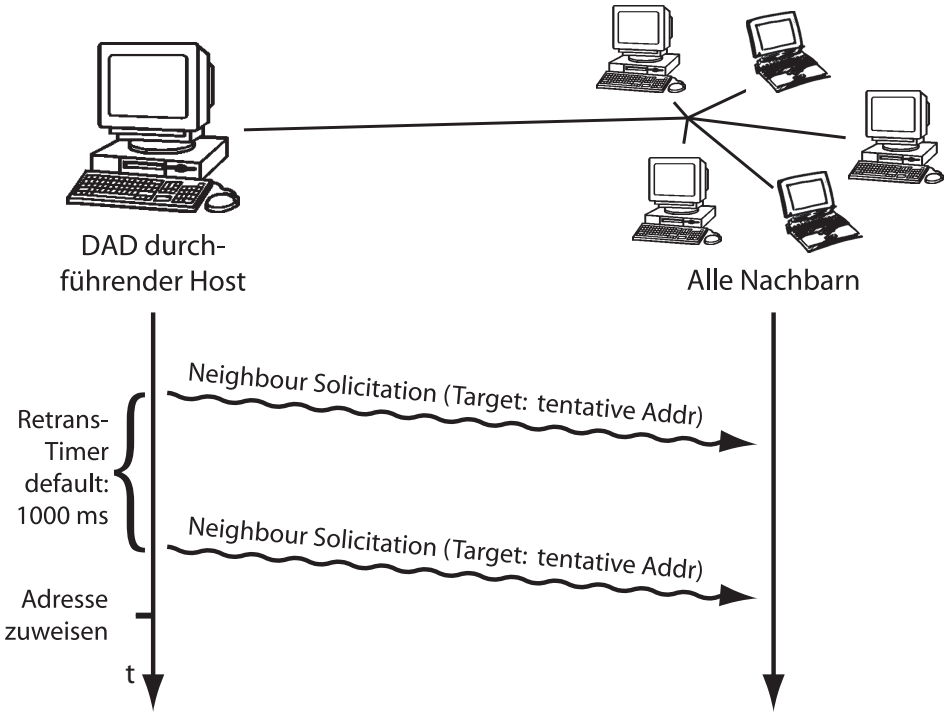

Abbildung 1: Ablauf: Duplicate Address Detection

Der Autokonfigurationsmechanismus will aber keinesfalls so lange warten, wie der Zeitabstand zwischen zwei Router-Advertisements in der Regel ist. Aus diesem Grund sendet der Host ein Router-Solicitation an die Multicast-Gruppe aller Router auf dem Link (ff02::2). Jeder Router hört auf diese Adresse und schickt beim Empfang einer solchen Nachricht ein Router-Advertisement an den anfragenden Host zurück.

Router-Advertisements enthalten Informationen, die einem Host erlauben, seine globalen Adressen sowohl zustandslos als auch zustandsbehaftet (DHCPv6) zu konfigurieren. Dazu enthalten die im Router-Advertisements enthaltenen Präfix-Informationen optional jeweils ein Flag, welches angibt, ob das entsprechende Präfix zur autonomen Adresskonfiguration verwendet werden soll, oder nicht. Die vom Router verschickten Präfixe betreffen die ersten 64 Bit der Adressen, mit denen sich die Hosts konfigurieren. Die letzten 64 Bit werden vom Host selber wie bei der Link-Lokal-Adresse generiert und an das vom Router verschickte Präfix angehängt:

```
<Präfix>:<erste Hälfte der MAC-Adresse>ff:fe<zweite Hälfte der MAC-Adresse>/64
```

Die Router-Advertisements enthalten noch zwei weitere Flags, die dem anfragenden Host mitteilen, welche Informationen er möglicherweise über zustandsbehaftete Adress-Autokonfiguration (DHCPv6) beziehen soll. Dabei wird mit den Flags zwischen den Informationen über die eigenen Adressen und andere Informationen (Nameserver, NTP-Server usw.) unterschieden.

Vor der endgültigen Zuweisung einer globalen Adresse an einem Interface, muss der Host für diese Adresse zur Sicherheit DAD ausführen. Das gilt sowohl für zustandsbehaftete als auch für zustandslose Autokonfiguration. Sogar die manuelle Konfiguration von IPv6 Adressen sollte mit DAD abgesichert werden. Die letzte Überprüfung kann allerdings von dem lokalen Netzwerkadministrator ausgeschaltet werden.

Um den Autokonfigurationsprozess zu beschleunigen, dürfen Hosts, währen sie eine Link-Lokal-Adresse konfigurieren und ihre Eindeutigkeit überprüfen, gleichzeitig auf ein Router-Advertisement zu warten. Da ein Router seine Antwort auf ein Router-Solicitation einige Sekunden verzögern kann, könnte die komplette Zeit für die Autokonfiguration wesentlich länger dauern, falls diese beiden Schritten hintereinander durchgeführt werden.

1.3 Probleme bei DAD

DAD wird benutzt, um IPv6 Adressen zustandslos zu konfigurieren - das heißt, dass man keine zentrale Instanz benötigt, die eine Übersicht über das Netzwerk aufrechterhält. Stattdessen verlässt sich DAD während dem Autokonfigurationsprozess auf die Kooperation der schon konfigurierten Knoten. Die Vorteile dieser Vorgehensweise sind leicht nachzuvollziehen: erstens muss man keinen zentralen Server konfigurieren, zweitens könnte keinen zentralen Zustand verloren gehen und drittens wird das Neustarten eines Router alle Adressen auf dem Subnetz nicht annullieren.

Leider ziehen diese Vorteile einige Nachteile mit sich – ein klarer Nachteil von DAD ist, dass der ganze Prozess auf Knoten beruht, die die eigene Adresse verteidigen. Es gibt keine positive Bestätigung, dass ein Knoten eine bestimmte Adresse tatsächlich verwenden darf. Vielmehr bedeutet dies, dass dieser Prozess sehr empfindlich auf Paketverluste ist – falls z.B. eine Signalisierungsnachricht verloren geht, könnte eine Adresskollision unentdeckt bleiben, was wiederum erhebliche Folgen hat. Wahrscheinlich der größte Nachteil von DAD ist aber die Zeit, die für eine Autokonfiguration vergeht und insbesondere das folgende Knotenverhalten: ein Knoten muss während dem Autokonfigurationsprozess für eine bestimmte Zeit auf eine negative Nachricht warten und ist nicht in der Lage eine positive Nachricht zu akzeptieren. So wäre der Knoten in der Lage eine Adresse sofort zu konfigurieren und diese ab sofort zu benutzen. Die Zeit, die das Protokoll braucht, um die Eindeutigkeit einer Adresse zu überprüfen spielt bei festen Knoten zwar keine große Rolle, sie ist aber bei mobilen Knoten von einer großen Bedeutung, da bei mobilen Applikationen für eine nahtlose Kommunikation möglichst kürzere Handoverzeiten benötigt werden.

Bei einer Adresskonfiguration entstehen durch DAD verschiedene Zeitverzögerungen. Die größte davon ist die 1000 ms Verzögerung, die für die wiederholte Versendung eines Neighbour Solicitations benötigt wird. Im nächsten Kapitel werden die vier verschiedenen Erweiterungen zu DAD vorgestellt. Ziel dieser Erweiterungen ist die Beschleunigung des Adresskonfigurationsprozesses.

2 DAD-Alternativen

Die zustandslose Adress-Autokonfiguration ist für kleinere Netze zwar sinnvoll, ist aber bei größeren Netzen nicht in der Lage alle Anforderungen zu erfüllen. Die wesentlichen Punkte wieso man weitere Protokolle benötigt sind folgende:

- bei zustandsloser Adress-Autokonfiguration kann die Adressvergabe kann nicht beeinflusst werden,
- es gibt keine zentrale Verwaltung von Adressen,
- man keine DNS-/NTP-Server ermitteln usw.

Das folgende Kapitel stellt die zurzeit existierenden Erweiterungen zu DAD vor. Alle vorgestellten Methoden sind zustandsbehaftet und sind von einer zentralen Instanz abhängig – ein

Server oder Router muss also positive Bestätigungen verteilen, dass eine Adresse tatsächlich frei ist und benutz werden darf. Deswegen sind alle Erweiterungen infrastrukturabhängig, was wiederum eine Auswirkung auf die Skalierbarkeit und die Ausfallsicherheit geben könnte.

2.1 Dynamic Host Configuration Protocol for IPv6

Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [DBVL+03] ist ein Protokoll, das dazu dient Netzwerkrechner dynamisch über einen zentralen Server zu konfigurieren. Die wesentlichen Aufgaben von DHCPv6 sind die Adressvergabe, die DNS-/NTP-Server-Adressübermittlung, Präfix Delegation und weitere. DHCPv6 verwendet UDP für den Nachrichtenaustausch. Der Klient verwendet immer seine Link-Lokal-Adresse, was den Vorteil hat, dass der Server dem Klient direkt antworten kann und kein Broadcast verwenden werden muss. Der Klient verwendet für die Kommunikation Port 546 und der Server – Port 547, außerdem sind Router über die folgenden Link-Lokal- und Site-Lokal-Multicast-Adressen zu erreichen:

```
ff02::1:2 (Alle DHCP-Relay-Agents & -Server; link-local)
ff05::1:3 (Alle DHCP-Server; site-local)
```

Neu bei DHCPv6 ist die Möglichkeit nicht nur Rechner im eigenen Subnetz zu konfiguriert, sondern auch Rechner, die sich in einem anderen Subnetz befinden. Damit nicht in jedem Linksegment ein eigener DHCPv6 Server stehen muss, werden die so genannten *Relay Agents* eingesetzt. Die Relay Agents nehmen die Anfragen von Klienten auf und leiten sie an die Unicast Adresse eines DHCPv6 Servers weiter. Der DHCPv6 Server antwortet dem Relay Agent und dieser wiederum antwortet dem Klient. Pro Linksegment sollte mindestens ein Relay Agent eingesetzt werden und zwischen einem Klient und einem Server können mehrere Relay Agents positioniert sein.

Im Gegensatz zu DHCPv4 wird bei DHCPv6 keine *Router-Option* zum Festlegen des Default Gateways verwendet. Sollte ein Default Gateway benötigt werden, muss dies in Router Advertisements übermittelt werden. Bei DHCPv6 können Daten zustandslos und zustandsbehaftet verwaltet werden. Zustandslose Daten können z.B. die feste IP eines DNS-/Zeitserver sein. Im Gegensatz bedeutet zustandsbehaftet die Zuteilung und Verwaltung von Ressourcen – z.B. die einmalige Vergabe einer IP Adresse. Dabei wird auf dem Server eine IP-Adresse einer MAC Adresse zugeordnet. Die IP Adresse kann sowohl dynamisch als auch statisch vergeben werden. Die Zuordnung zwischen IP Adresse und Host wird durch ein *DHCP Unique Identifier (DUID)* und ein *Identity Association Identifier (IAID)* erreicht. Die DUID kennzeichnet einen Client oder Server und ist dabei nicht von der Hardware abhängig. Das heißt, dass sich die DUID nicht ändert wenn Hardware ausgetauscht wird. Eine IAID kennzeichnet eins von vielen möglichen Interfaces.

Das Protokoll besitzt einen Basisheader (siehe Abbildung 2), in welchem Optionen variabler Länge eingebunden sein können. Grundsätzlich enthalten sind der *msg-type* sowie eine *transaction-id*. Die msg-type kennzeichnet den Typ der Nachricht z.B. *SOLICIT*, *ADVERTISE*, *REQUEST*, *REPLY* usw. Die transaction-id ist eine Nummer, die den Zusammenhang zwischen einer Anfrage und einer Antwort herstellt. Die transaction-ids von einem *SOLICIT<->ADVERTISE*- sowie einem *REQUEST<->REPLY*-Paar müssen also gleich sein.

Bei DHCPv6 gibt es, genauso wie bei DHCPv4, Sicherheitsprobleme. Die 3 wichtigsten sind:

- Unbekannte externe DHCPv6 Server, welche den Klienten falsche Adressen zuweisen

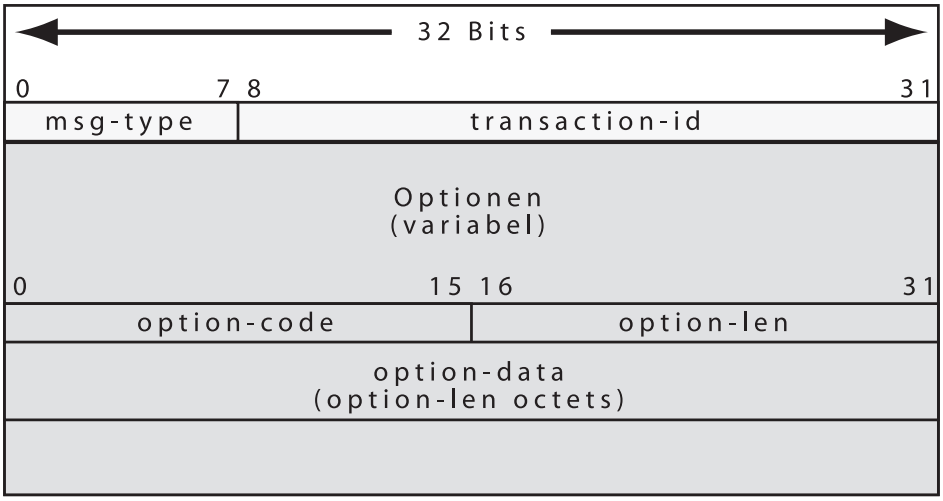

Abbildung 2: DHCPv6-Paket (Server & Klient)

- Unsachgemäß oder böswillig aufgesetzte DHCP Server im internen Netz, die die Klienten konfigurieren.
- Fremde Klienten, die sich in das interne Netz hängen und automatisch eine interne Adresse erhalten.

Schutz vor diesen Angriffen bietet der Einsatz von *DHCP Authentication*. Mit Hilfe einer Authentication Option kann die Herkunft einer DHCPv6 Nachricht eindeutig identifiziert werden. Außerdem ist somit sichergestellt, dass der Inhalt der Nachricht unterwegs nicht verändert wurde.

Die zustandbehaftete, Server basierte Host-Konfiguration, die DHCPv6 bietet, löst ein Teil der durch DAD entstandenen Probleme. Allerdings verbleibt die Adresskonfigurationsverzögerung, die für mobile Geräte von einer sehr großen Bedeutung ist. Der Grund dafür ist, dass DHCPv6 über die Link-Lokal-Adresse mit einem Host kommuniziert – diese muss also schon konfiguriert werden. Das bedeutet wiederum, dass die RetransTimer Verzögerung von 1000 MIllisekunden verbleibt. Vielmehr verlangen IPv6-Standards, dass alle mit DHCPv6 vergebenen Adressen zusätzlich mit DAD überprüft werden müssen, weil ein DHCPv6 Server kein Wissen über die zustandslos konfigurierten Hosts hat. Das würde also die gesamte DAD Verzögerung zusätzlich erhöhen, weil zuerst dir Link-Lokal-Adresse und dann die globale Adresse mit DAD überprüft werden müssen, anstatt beide parallel durchzuführen. Der zweite Nachteil von DHCPv6 ist der zentrale Server-abhängige Netzwerk-Zustand, der die Netzwerk-Ausfallsicherheit verringert. DHCPv6 ist zwar eine sehr elegante Lösung für die Zuteilung von Host-Konfigurationsinformationen, ist aber dennoch, aufgrund der vorgestellten Nachteile, keine angemessene Alternative von DAD, was die Konfigurationsverzögerung betrifft.

2.2 Advance Duplicate Address Detection (ADAD)

Advance Duplicate Address Detection (ADAD) [HCJP03] ist ein Protokoll, das entwickelt wurde, um den großen Anforderungen der mobilen Technologien entgegenzukommen. Da mobile Knoten bei einem Linkwechsel schnelle Konfiguration einer neuen Care-Of-Adresse benötigen, ist es zwingend notwendig die durch DAD entstehende Verzögerung für die Adresskonfiguration zu reduzieren.

ADAD versucht sogar die durch DAD entstandene Verzögerung komplett zu eliminieren. Das geschieht indem neue Care-Of-Adressen im Voraus konfiguriert werden und später nach dem Linkwechsel eines mobilen Knotens sofort verwendet werden können. Dafür müssen Access Router genügend viele Adressen im Voraus generieren und alle diese Adressen müssen dann mit dem üblichen DAD überprüft werden. Alle Adressen, die eindeutig sind, kommen dann in einen so genannten *Passive Proxy Cache*.

Die nachfolgende Konfigurationsprozedur seitens des mobilen Knotens unterscheidet zwischen zwei Fälle: prädiktive und nicht-prädiktive Adresszuweisung. Im prädiktiven Fall bekommt der mobile Knoten eine neue Care-Of-Adresse, aus den eindeutig vorkonfigurierten Adressen, bevor er sein Link wechselt. Der mobile Knoten bekommt die neue Adresse von seinem aktuellen Access Router, der seinerseits diese Adresse mit Hilfe spezieller Handover-Signalisierung vom neuen Access Router empfängt. Im nicht-prädiktiven Fall findet keine Adresskonfiguration vor dem Linkwechsel statt. Stattdessen bekommt ein mobiler Knoten sofort nach der Verbindungsherstellung, auf dem neuen Link, eine eindeutige Adresse, die er sofort benutzen darf, ohne keinerlei DAD Überprüfungen bzw. ohne Verzögerung.

Die drei wichtigsten Vorteile dieses Protokolls sind folgende:

- die DAD-Verzögerung für die Konfiguration der globalen Adresse wird komplett eliminiert
- eine Adresskollision ist völlig ausgeschlossen (falls es keine Paketverluste gibt)
- da die neue Adresse sicher eindeutig ist, kann sie sofort in Nachrichten als Source-Adresse benutzt werden

Als Nachteil könnte man den Fakt bezeichnen, dass der mobile Knoten kein Einfluss auf die neue Adresse hat, sozusagen keine Wahlmöglichkeit. Das spielt eine bedeutende Rolle z.B. bei Secure Neighbour Discovery. Dieses Vorgehen beruht auf kryptografisch generierten Adressen (vom eigenen privaten Schlüssel). Demzufolge sind Knoten, die von ADAD abhängig sind, aus Secure Neighbour Discovery ausgeschlossen.

2.3 Multicast Listener Discovery DAD (MLD-DAD)

Wenn ein Host die übliche DAD Prozedur abschließt, tritt er zunächst die *Solicited-Nodes multicast Gruppe* ein und muss später auf Nachrichten hören, die an diese Gruppe adressiert sind. So ist der Knoten in der Lage Advertisements, adressiert an diese Gruppe, zu empfangen und eventuelle Adresskollisionen zu entdecken. Wenn also im Netzwerk eine Instanz gäbe, die die Übersicht behält, welche Solicited-Nodes Adressen verwendet werden, wüsste diese Instanz auch welche unicast Adressen zur Verfügung stehen.

Der *Multicast Listener Discovery (MLD)* Standard schreibt einem Host vor, einen MLD-Bericht zu senden, bevor er anfängt auf die *Solicited-Node multicast Adresse* zu hören (wie sich eine solche Adressen bildet, ist in Abbildung 3 dargestellt). Das bedeutet, dass jeder Host den Access Router auf dem Link über die eigene Anwesenheit informieren muss. So weiß der Router, welche „Hörer“ es auf dem Link gibt und die entsprechenden multicast Gruppen können verwaltet werden. Dieser Standard ist die Grundlage für die *MLD Optimierung für DAD (MLD-DAD)* [**?**]. Dieses Protokoll erlaubt einem Knoten den zuständigen Access Router zu erfragen, ob er der erste Knoten ist, der eine bestimmte multicast Gruppe beitreten will. Falls dies der Fall sein sollte, bedeutet dies, dass kein anderer Knoten die gleiche Adresse besitzt und diese könnte dann sofort an dem Interface zugewiesen werden. Das Protokoll ist unten genauer vorgestellt.

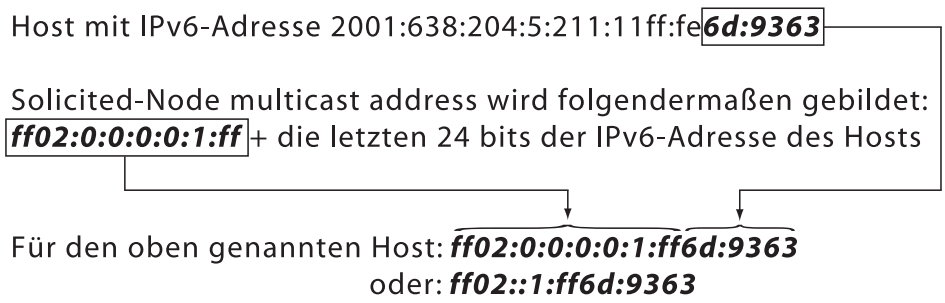

Abbildung 3: Bildung einer Solicited-Node multicas Adresse

Wenn ein Host eine multicast Gruppe eintritt, um eine DAD Überprüfung durchzuführen, sendet er einen MLD-Bericht an die *link scope solicited-nodes multicast Adresse*. Dieser Bericht muss vor dem von DAD vorgeschriebenen Neighbour Solicitation gesendet werden und verlangt eine Antwort (Report-Requesting-Response), falls die Adresse, die gerade berichtet wird, zur erfragten multicast Gruppe gehört. DAD wird nebenbei ganz normal durchgeführt, kann aber unterbrochen werden, falls eine der folgenden drei Ereignissen zustoßt:

- DAD abläuft – das bedeutet, dass keine doppelte Adresse gefunden wurde
- Ein Neighbour Solicitation oder Advertisement für die tentative Adresse wurde empfangen – das bedeutet, dass die Adresse verwendet wird und sie muss dekonfiguriert werden
- Eine MLD-Antwort wurde empfangen, die darauf hinweißt, dass keine andere „Hörer“ für diese multicast Gruppe existieren - das bedeutet, dass die Adresse zugewiesen und benutzt werden darf

Ein Router, der MLD-Antworten bereitstellt, muss entsprechend MLD-Berichte bearbeiten und Informationen über existierende multicast Gruppen verwalten. Wenn ein solcher Router einen MLD-Bericht empfängt, der eine Antwort verlangt, muss der Router zuerst überprüfen, ob dieselbe multicast Gruppe überhaupt auf dem Link existiert. Falls die Gruppe nicht existiert wird sie erstellt und nachdem der Status der multicast Gruppe aktualisiert wird, sendet der Router eine Antwort an die multicast Adresse dieser Gruppe. So wird dem Host mitgeteilt, dass er der erste ist, der in diese Gruppe eintritt.

Wenn ein MLD-Router neu gestartet wird, muss er zuerst Wissen über alle multicast Gruppen, die auf dem Link existieren bekommen. Das geschieht anhand *General Query Messages*, die der Router jedes *Query Interval* sendet. Solange der MLD-Router kein komplettes Wissen über das Netzwerk hat, darf er keine MLD-Berichte beantworten.

Falls auf dem Link keine Router existieren, können entsprechend MLD-Berichte nicht beantwortet werden. In diesem Fall wird DAD ganz normal durchgeführt. Falls zwei Hosts gleichzeitig einen MLD-Bericht abschicken, für welche noch keine multicast Gruppe existiert, beantwortet der Router nur den ersten Bericht, der er empfängt. Der zweite Host muss also DAD normal durchführen. Es besteht leider die Möglichkeit, dass der erste Host die MLD-Antwort erst nach dem Neighbour Solicitation des zweiten Routers empfängt, was dazu führt, dass er die tentative Adresse dekonfiguriert und im Endeffekt keiner der beiden Hosts eine Adresse erfolgreich konfiguriert.

Schließlich ist MLD-DAD ein Protokoll, das die typische DAD Verzögerung umgeht. Dieses Protokoll ist eine gute Alternative, die allerdings sehr infrastrukturspezifisch ist. Es ist zwingend notwendig, dass komplett alle Knoten im Subnetz MLD korrekt durchführen, sonst ist

die Verfügbarkeit einer Adresse nicht mehr sicher. Aber in einem Netzwerk, wo alle Knoten die entsprechenden Anforderungen erfüllen ist MLD-DAD eine schnelle und effiziente Alternative zu DAD.

2.4 Optimistic Duplicate Address Detection (ODAD)

Die letzte und aktuellste DAD Alternative ist das *Optimistic Duplicate Adress Detection (ODAD)* [Moor05]. Dieses Protokoll ist eine Modifikation der existierenden IPv6 Neighbour Discovery Mechanismen und des zustandslosen Adress-Autokonfigurationsprozesses. Ziel des Protokolls ist die Adresskonfigurationsverzögerung im erfolgreichen Fall zu reduzieren und eventuelle Störungen bei einem Fehlschlag möglichst zu vermeiden. Dieses Protokoll ist eine interessante und viel versprechende Alternative, weil das übliche DAD eine komplett pessimistische Strategie darstellt. Das bedeutet, dass solange sich ein Interface nicht vergewissern hat, dass seine tentative Adresse tatsächlich eindeutig ist, sind alle Kommunikationen auf diesem Interface gesperrt. In der Regel ist aber sehr Wahrscheinlich, dass DAD erfolgreich durchläuft, insbesondere wenn die Adressen sorgfältig ausgesucht werden.

Die existierende IPv6 Adresskonfigurationsmechanismen bieten einen sehr guten Schutz von Adresskollisionen, indem diese rechtzeitig entdeckt werden. Allerdings gibt es immer mehr mobile Geräte, die häufig das Subnetz wechseln und für welche eine schnellere Adresskonfiguration unabdingbar ist. Die ziele die sich ODAD setzt sind die folgenden Optimierungen:

- Kompatibilität mit Knoten des aktuellen Standards (RFC 2462).
- Eliminierung der RetransTimer-Verzögerung während Adresskonfiguration.
- Sich vergewissern, dass die Kollisionswahrscheinlichkeit nicht erhöht wird.
- Verbesserung der Konfliktauflösungsmechanismen im Falle einer Adresskollision.
- Minimierung der Störungen bei einer Adresskollision.

ODAD ändert die Regeln, beschrieben in RFC 2462, ab und erlaubt einem Knoten mit der Welt über eine tentative Adresse zu kommunizieren. Allerdings versucht man die möglichen Störungen bei einer doppelt konfigurierten Adresse zu minimieren. Diese Änderungen erlauben die problemlose Kommunikation zwischen normalen und optimistischen Knoten. Ein Knoten der Pakete senden oder empfangen will muss allerdings Neighbour Discovery durchführen bzw. teilnehmen. Das Risiko bei einer doppelten Adresse besteht in der Möglichkeit, dass Neighbour Solicitations, Neighbour Advertisements oder Router Solicitations, gesendet von einer tentativen Adresse, eine widrige Auswirkung auf anderen Knoten haben: z.B. könnten Einträge in Neighbour Caches falsch geändert werden usw.

Um diese Probleme umzugehen nutzt ODAD die existierenden Flags und Optionen in Neighbour Discovery Nachrichten aus. Neighbour Advertisements werden hierfür mit gelöschtem *Override Bit* gesendet. Dagegen werden Neighbour Solicitations und Router Solicitations ohne Source Link-Layer Adresse Option gesendet. Ein ODAD Knoten muss noch seinen Neighbour Discovery Mechanismus leicht abändern und an den Anforderungen anpassen – ein solcher Knoten muss z.B. Pakete für unbekannte Ziele nur über einen Router senden und kein Neighbour Discovery anstoßen. Die vorgestellten Einschränkungen vermeiden eine fälschliche Beeinflussung der Neighbour Cache Einträge durch eine doppelte tentative Adresse. Die Neighbour Discovery Mechanismen sind zwar nicht so effizient solange eine Adresse tentative ist, aber nach dem Ablauf der DAD-Prozedur ist diese Adresse nicht mehr tentative und es gilt wieder das übliche Neighbour Discovery Verhalten.

Die Wahrscheinlichkeit, dass ein ODAD Knoten trotz einer Adresskollision fälschlicherweise kommunizieren würde, ist sehr gering. Aufgrund des gelöschten Override Bits werden ihm seine Nachbarn nicht erlauben, das Neighbour Discovery abzuschließen und sobald der Knoten das verteidigende Neighbour Advertisement empfängt, muss er diese doppelte Adresse sofort dekonfigurieren.

ODAD ist eine versprechende DAD-Erweiterung, die besonders effizient für Netzwerke ist, wo der Versand einiger zusätzlicher Nachrichten pro konfigurierenden Knoten kein großer Mehraufwand ist. Zusätzlich, aufgrund der reduzierten Störungen bei einer Adresskollision, könnte ein Knoten mehrfach untersuchen ob eine Adresse schon vergeben ist. Das verringert sehr die Wahrscheinlichkeit, dass aufgrund Paketverluste eine Adresskollision unentdeckt bleibt, was wiederum ODAD besonders passend für Drahtlose Netzwerke macht.

3 Fazit

Ziel dieser Seminararbeit war die Adress-Konfiguration in IPv6 und die existierenden Optimierungen vorzustellen. Die einzelnen Protokolle wurden beschrieben und die jeweiligen Vor- und Nachteile wurden angegeben. Als größter Nachteil der Adress-Konfiguration hat sich die Verzögerung ergeben, die durch die Überprüfung entsteht, ob eine Adresse im Netz schon verwendet wird oder nicht. Während dieser Überprüfung ist ein Knoten nicht in der Lage mit der Welt zu kommunizieren, was von besonderer Bedeutung für mobile Geräte ist. Alle vorgestellte Optimierungen versuchen diese Verzögerung umzugehen oder sogar komplett zu eliminieren. DHCPv6 bietet zwar eine sehr fortgeschrittene Verwaltung und Zuteilung von Netzwerk Informationen, ist aber nicht in der Lage die wichtige Verzögerung für die Adresskonfiguration umzugehen. Dagegen sind ADAD und MLD-DAD zwei Protokolle, die die Adresskonfigurationsverzögerung sehr gut umgehen, allerdings sind diese Protokolle Infrastrukturabhängig, was die Skalierbarkeit verschlechtert. Zuletzt wurde ODAD vorgestellt, das sehr elegant die Verzögerung umgeht und dabei große Kompatibilität zur existierenden Standards hat. Dieses Protokoll ist für mobile Geräte, die das Subnetz häufig wechseln eine sehr gut geeignete Lösung.

Literatur

[DaNe03] Greg Daley und Richard Nelson. Duplicate Address Detection Optimization using IPv6 Multicast Listener Discovery. Internet Draft, Februar 2003.

[DBVL+03] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins und M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC3315 (Standards Track), Juli 2003.

[HCJP03] Y. Han, J. Choi, H. Jang und S. Park. Advance Duplicate Address Detection. Internet Draft, Juli 2003.

[Moor05] Nick Moore. Optimistic Duplicate Address Detection for IPv6. Internet Draft, Dezember 2005.

[ThNJ05] S. Thomson, T. Narten und T. Jinmei. IPv6 Stateless Address Autoconfiguration. Internet Draft, Mai 2005.

Abbildungsverzeichnis

Flow Movement mit Mobile IPv6

Rolland Veres

Kurzfassung

Die moderne Informationsgesellschaft ist in hohem Masse auf des Internet angewiesen. Sei es beruflich oder privat, ohne Zugang zum großen Netzwerk, bleibt einem eine wichtige Ressource verschlossen. Für Desktop-Rechner ist solch ein Zugang jedoch schon längst zum Standard geworden. Neue Herausforderungen bringen aber kleine und mobile Geräte, die immer mehr Einzug in unser Leben halten. Gerade durch die Mobilität fällt es schwer, ständige Verbindung mit dem Internet zu halten. Zwar gibt es eine ganze Reihe verschiedener drahtloser Netztechnologien, doch je größer die Reichweite, desto geringer sind die Datenraten und desto höher sind die Zugangskosten. Für den Benutzer dieser Technologien kann es dabei schwierig werden, den Überblick zu behalten, welche Zugangstechnologien verfügbar und für die eigenen Zwecke am besten geeignet sind. Deshalb wird die Thematik der Verwaltung von Datenströmen und der dazugehörigen Netwerkschnittstellen immer wichtiger. In dieser Arbeit wird auf diese Thematik eingegangen und geprüft, welche Vorraussetzungen für ein funktionierendes Flow Movement im Rahmen des Mobile-IPv6-Netzwerkprotokolls bereits erfüllt sind, und welche Erweiterungen desselben nötig sind.

1 Einleitung

Vernetzung ist in der heutigen Informationsgesellschaft ein immer wichtigerer Punkt der sogar zum Standard wird. Im Heimbereich sind längst die meisten Haushalte mit dem weltweiten Netz verbunden, doch auch in öffentlichen Bereichen bekommt man immer häufiger die Möglichkeit “Online” zu gehen, jedoch muss das eigene Gerät, mit der entsprechenden Netzwerktechnologie ausgestattet sein. Zudem bieten diese unterschiedlichen Netze auch unterschiedliche Attribute, die nicht immer optimal für die gewünschte Anwendung sind. So unterscheiden sich, zum Beispiel, WLAN (802.11), Bluetooth und GPRS enorm in ihren Eigenschaften, was Reichweite, Datenkapazität und meist auch Kosten angeht. Hinzukommt, dass nicht immer gewährleistet ist, dass ein spezielles Netz verfügbar ist. Daher ist es immer häufiger der Fall, dass Geräte, wie Notebooks, PDAs oder Ähnliches, mit gleich mehreren Schnittstellen für diese verschiedenen Technologien ausgestattet sind. Die Mobilität heutiger Geräte machte es notwendig, Netzwerkprotokolle so zu erweitern, dass diese Mobilität auch tatsächlich nutzbar ist und dies möglichst komfortabel. Das folgende Beispiel soll die Problematik aufzeigen und den Leser mit den Elementen, der in dieser Arbeit behandelten Themen, vertraut machen.

Ein Geschäftsmann ist vom Bahnhof in die Innenstadt unterwegs. Bei sich hat er einen Tablet-PC mit den drei Netzwerkschnittstellen für WLAN (802.11), Bluetooth 2.0 und GPRS. Am Bahnhof steht ein WLAN-Zugriffsknoten, über den der Tablet-PC verbunden ist. Über diese Verbindung läuft ein Datendownload sowie wichtige Nachrichten eines Web-Radios. Der Geschäftsmann beschließt ein Telefonat per VoIP zu machen. Dieses soll auch bei Verlust der WLAN-Verbindung nicht beeinträchtigt werden und wird deshalb über eine neue GPRS-Verbindung initiiert. Nach kurzer Zeit verlässt der Geschäftsmann den Bahnhof und die

Signalqualität des WLAN degradiert sehr stark. Dafür kommt der WLAN-Zugriffsknoten eines großen Kaufhauses in Sicht. Es wird eine Übergabe vom WLAN-Subnetz des Bahnhofes an das WLAN-Subnetz des Kaufhauses eingeleitet, welcher jedoch eine Zeit benötigt, in der der Datenstrom über diese Schnittstelle unterbrochen ist. Damit das Web-Radio keinen hörbaren Aussetzer hat, wird es kurzfristig auf die GPRS-Verbindung umgeleitet, da dort noch Kapazitäten neben dem Anruf vorhanden sind. Bei dem Datendownload kommt es kurzfristig zu einer Unterbrechung, welche aber dem Geschäftsmann nicht negativ auffällt. Noch bevor die Verbindung mit dem WLAN des Kaufhauses aufgebaut ist, kommt ein Bluetooth-Zugriffspunkt in Sichtweite. Die Verbindung wird aufgebaut, und um die teure GPRS-Volumen-Flatrate zu entlasten, wird das Web-Radio auf diese Verbindung umgeleitet. Nachdem die Verbindung mit dem WLAN des Kaufhauses steht, ist die Signalqualität leider so schlecht, dass auch der Datendownload auf die Bluetooth-Schnittstelle umgeleitet wird. Nachdem der VoIP-Anruf beendet ist, benötigt der Geschäftsmann nun dringend den Datendownload, und da die WLAN-Verbindung ganz weggefallen ist und auch die Bluetooth-Verbindung schwächer wird, leitet der Geschäftsmann den Datendownload auf GPRS um, und unterbricht das Web-Radio, nachdem die Bluetooth-Verbindung ganz abgebrochen ist.

Thema dieser Arbeit ist die Verwaltung von Datenströmen, wie zum Beispiel Datendownloads, Web-Radio oder ein VoIP-Gespräch, wie sie im Beispiel aufgetaucht sind, und den zugeordneten Netzwerkschnittstellen im Mobile-IPv6-Netzwerkprotokoll. Das Ziel dabei ist, dem Benutzer, möglichst transparent, immer die beste Verbindung zu bieten. Nachfolgend soll diese Thematik genauer betrachtet werden. Zunächst wird auf einige Grundlagen eingegangen, inklusive der dazugehörigen Terminologie, anschließend werden zwei Arten des Flow Movements erläutert. Danach werden einige Erweiterungen des Mobile-IPv6-Protokolls vorgestellt, die gewisse Aspekte des Flow Movements ermöglichen sollen.

2 Grundlagen und Terminologie

Bevor detailliert auf das Flow Movement und dessen Realisierung eingegangen wird, sollen in diesem Kapitel zunächst einige Grundlagen vermittelt und Terminologien erläutert werden.

Um zu verstehen, wozu Flow Movement nötig ist, muss zunächst geklärt werden, was ein Flow ist und worin das Movement besteht. Mit Flow ist ein Datenfluss, beziehungsweise eine Datenverbindung gemeint. Diese Verbindung besteht zwischen zwei Punkten im Netz und über die Verbindung werden Daten gesendet. Ein einfaches Beispiel für die Öffnung einer solchen Datenverbindung, ist der Abruf einer Webseite im Internet. Die Endpunkte sind der Rechner, auf dem der Browser läuft, und der Server, auf dem die Daten der Webseite liegen. Im Beispiel der Einleitung waren drei andere solcher Datenverbindungen beschrieben: ein Datendownload, der Datenstrom eines Web-Radios und ein VoIP-Anruf. Für jeden Datenstrom wird eine neue Datenverbindung geöffnet. Es kommt schnell vor, dass man mehrere Flows mit seinem Gerät hält. Spätestens, wenn am Gerät verschiedene Netzwerkschnittstellen gleichzeitig aktiv sind, muss man sich Gedanken machen, welcher Flow über welche Schnittstelle läuft. In dieser Arbeit geht die Verwaltung der Datenverbindungen aber noch weiter, denn um das Szenario aus der Einführung realisieren zu können, müssen Flows, unter bestimmten Umständen, auch die Netzwerkschnittstelle wechseln können. Dieses Flow Movement soll in dieser Arbeit als Teil des Netzwerkprotokolls Mobile IPv6 betrachtet werden. Hierzu muss also zunächst geklärt werden, wie eine Datenverbindung aussieht, und wie die Änderung einer bestehenden Datenverbindung funktioniert. Mobile IPv6 ist ein Netzwerkprotokoll speziell für mobile Geräte, welche die Besonderheit haben, dass sie sich außerhalb ihres Heimnetzes aufhalten können und zudem, während einer Datenverbindung, eine bestehende Netzwerkverbindung verlieren oder eine neue Netzwerkverbindung aufbauen können. Damit dabei bestehende Datenverbindungen möglichst unbeeinflusst bleiben, gibt es zwei verschiedene Kommunikationsmodi.

- Route Optimization: Es besteht eine direkte Verbindung zwischen dem mobilen Knoten und dem Kommunikationspartner.
- Bidirektionales Tunneln: Das Mobile-IPv6-Protokoll ermöglicht es dem mobilen Knoten, sich weiterhin virtuell im Heimnetz zu befinden. Das so genannte Binding ordnet der Heimatadresse des mobilen Knotens eine Care-Of-Adresse zu, welche die Adresse des mobilen Knotens im Fremdnetz darstellt. Dieses Binding liegt auf dem Heimatagenten des mobilen Knotens. Besteht nun eine Datenverbindung zwischen dem mobilen Knoten und einem Kommunikationspartner, so schickt der mobile Knoten seine Daten durch den Heimatagenten an den Kommunikationspartner, dieser wiederum sendet Daten an die Heimatadresse des mobilen Knotens, und der Heimatagent übernimmt das Tunneln an die Care-Of-Adresse des mobilen Knotens.

Auch bei der Route Optimization existiert ein Binding, jedoch direkt auf dem Kommunikationspartner. Dieses Binding verbindet wiederum die Heimatadresse des mobilen Knotens mit seiner Care-Of-Adresse. Ändert sich nun die Care-Of-Adresse für eine Datenverbindung, so muss ein Binding Update verschickt werden. Dabei wird das neue Binding beim Heimatagenten registriert und nach der Antwortprozedur an den betreffenden Kommunikationspartner gesendet, falls Route Optimization betrieben wird. Abbildung 1 zeigt diese Zusammenhänge.

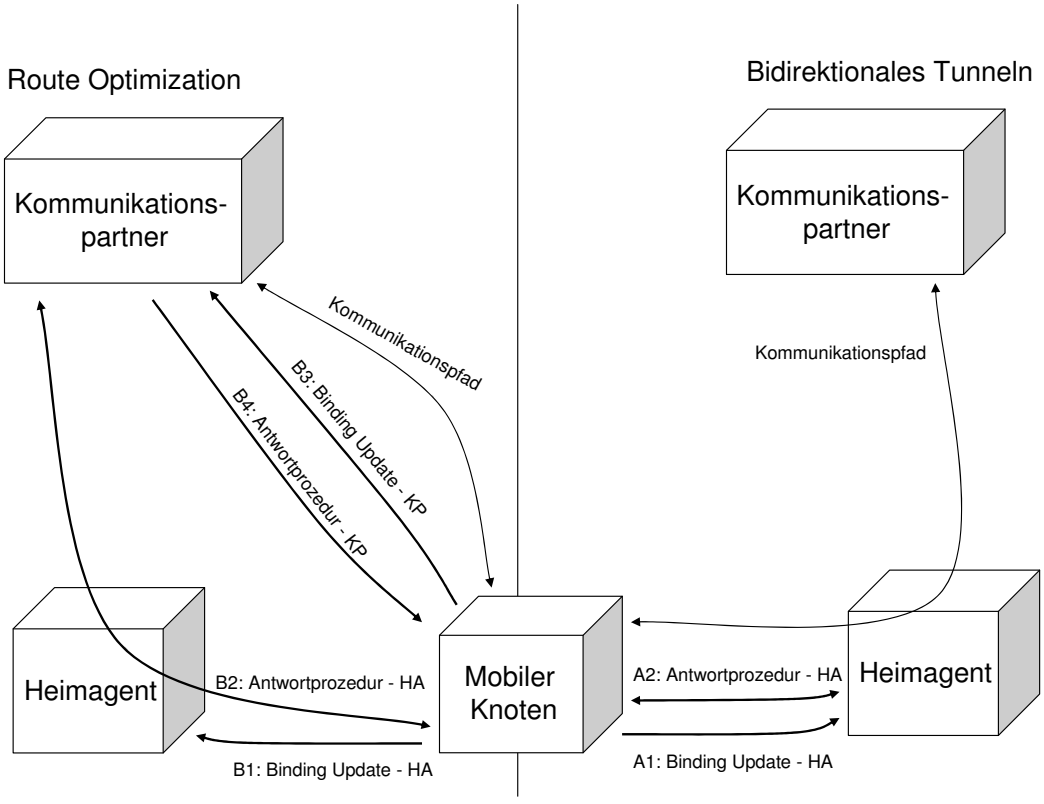

Abbildung 1: Binding Update bei Route Optimization und bidirektionalem Tunneln

Ein weiterer Begriff ist Multihoming. Dieser bedeutet, dass ein mobiler Knoten über mehrere IP-Adressen erreichbar ist. Dies kann auf verschiedenen Wegen erzielt werden. Hat ein mobiler Knoten mehrere aktive Schnittstellen, so wird er für jede eine IP-Adresse haben,

die er als Care-Of-Adressen für Kommunikationspartner und Heimatagent mit seiner Heimatadresse verknüpft. Hat ein mobiler Knoten mehrere Heimatagenten, so benötigt er für jeden eine eigene Heimatadresse, aus denen er wählen kann. Schließlich kann der mobile Knoten noch mehrere Care-Of-Adressen haben, eine für jedes Subnetz, in dem er sich befindet. Die Besonderheit an diesen drei Werten ist, dass sie unabhängig voneinander sind. Aus der Kombination dieser drei Parameter ergeben sich einige Szenarien, welche in [EMWP+05] und [MWEN+05], inklusive ihrer Auswirkungen, ausführlich beschrieben sind. Diese müssen beim Flow Movement berücksichtigt werden, wobei auch zu beachten ist, dass Flow Movement erst dann nötig wird, wenn Multihoming stattfindet, dieses entsteht nämlich erst, wenn der mobile Knoten tatsächlich über mindestens zwei Verbindungen in Netzwerke verfügt.

Die Änderung eines Binding für einen Datenfluss wurde bereits als wichtiger Aspekt des Flow Movements genannt. Dieser Vorgang wird im folgenden Abschnitt 3 weiter vertieft. An dieser Stelle sollen statt dessen, die Eigenschaften einer solchen Veränderung erläutert werden. Wenn sich das Binding einer Datenverbindung ändert, kann dies entweder plötzlich eintreten, oder vorhersehbar sein. Ist sie vorhersehbar, so ist es theoretisch möglich, den Übergang unterbrechungsfrei zu gestalten, anderenfalls spricht man von einem reaktiven Übergang, welcher zwangsweise zu einer Unterbrechung des Datenstroms führt, bis beide Seiten die Unterbrechung bemerkt haben, und eine neue Verbindung aufgebaut ist. Unabhängig davon, unterscheidet man beim Auftreten einer Unterbrechung in der Netzwerkverbindung, zwischen einem transparenten und einem nicht transparentem Übergang. Dies bezieht sich auf die Sichtbarkeit des Übergangs oberhalb von Schicht 3. Beim Flow Movement ist man bestrebt, Übergänge möglichst unterbrechungsfrei und transparent zu gestalten.

3 Verschieden Arten von Flow Movement

Betrachtet man mobile Geräte mit mehreren Netzwerkschnittstellen genauer, stellt man fest, dass es zwei Arten von Veränderung bei der Zuordnung eines Flows und dessen Netzwerkverbindung gibt. Man spricht dabei von einer Übergabe des Flows von einem Binding zu einem anderen. Die beiden Arten von Übergaben werden nachfolgend detaillierter vorgestellt.

3.1 Horizontale Übergaben

Ein mobiles Gerät hat den großen Vorteil, dass man nicht an Ort und Stelle gebunden ist, sondern einen großen Bewegungsfreiraum hat. Dadurch kann das Gerät leicht den eingeschränkten Bereich einer drahtlosen Netzwerkverbindung verlassen. Bei Computermessen beispielsweise, möchte der Veranstalter garantieren, dass die gesamte Messefläche mit WLAN abgedeckt ist, jedoch reicht ein Zugangspunkt bei weitem nicht aus, um dies zu bewerkstelligen. Die Lösung hierfür ist, mehrere Zugangspunkte so zu verteilen, dass die Bereiche sich leicht überlappen. Befindet sich ein mobiles Gerät in dem Einflussbereich von mehreren Zugangspunkt der gleichen Funkzugangstechnologie, kann dort eine horizontale Übergabe stattfinden (siehe Abbildung 2).

Vor einer solche Übergabe erfolgt zunächst die Anmeldung an dem neuen Zugangspunkt, bei dem man eine neue Care-Of-Adresse bekommt. Mit dieser neuen Care-Of-Adresse wird dann ein Binding Update durchgeführt, so dass jeder Flow, welcher die alte Care-Of-Adresse nutzte, das neue Binding nutzen kann. Hauptgrund für eine horizontale Übergabe ist, dass der aktuelle Zugangspunkt schlechtere Qualität bietet als ein anderer. Dies kann daran liegen, dass die Qualität des alten Zugangspunktes abnimmt und/oder die Qualität des neuen Zugangspunktes zunimmt. Häufig ist dies ein Anzeichen dafür, dass man sich im Überlappungsbereich eines größeren Netzwerkes befindet und daher sowieso in Kürze ein Subnetzwechsel erfolgen muss.

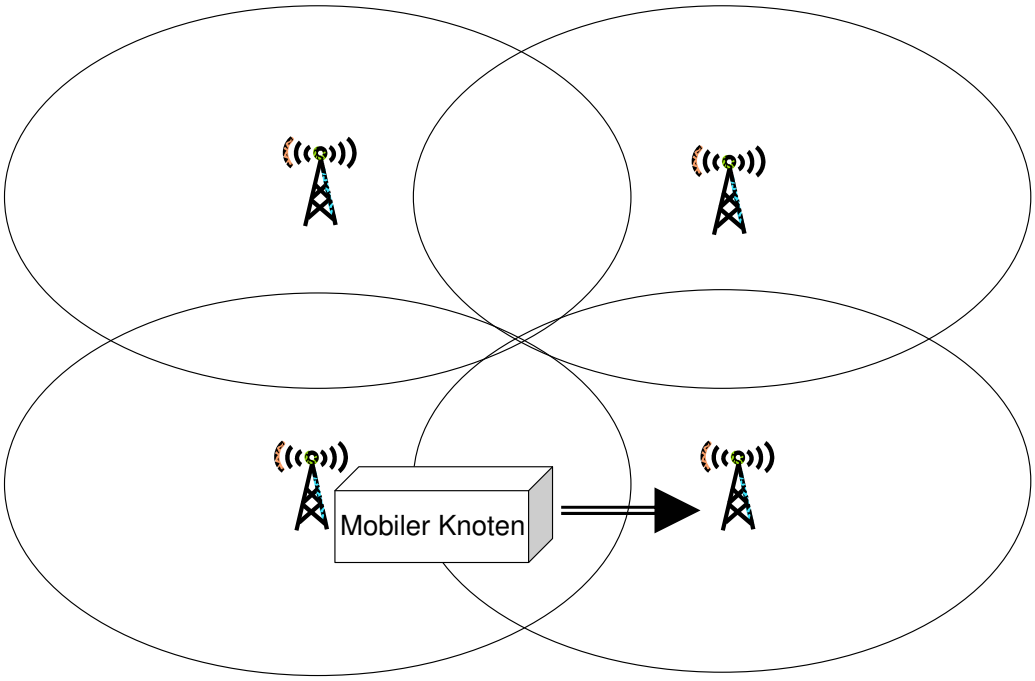

Abbildung 2: Abdeckungsbereich mehrerer Zugangspunkte der gleichen Technologie mit horizontaler Übergabe

3.2 Vertikale Übergaben

Jede Netzwerktechnik hat, wie bereits erwähnt, unterschiedliche Vor- und Nachteile, sowie spezielle Leistungsdaten. Des weiteren ist nicht immer genau die Art von drahtlosem Netzwerk vorhanden, welches einem am besten zusagt. Tatsächlich ist es mittlerweile so, dass man fast überall mindestens ein Netzwerk vorfindet. Insbesondere Mobilfunknetze sind so weit verbreitet, dass es fast immer möglich ist, über diese eine Netzwerkverbindung zu bekommen. Allerdings sind solche Verbindungen meist teuer und die Übertragungsraten nicht besonders groß. Es gibt jedoch auch andere Schnittstellen, die immer mehr Verbreitung finden. Daher ist es nicht überraschend, dass auch immer mehr dieser Schnittstellen gleichzeitig in mobilen Geräten vorhanden sind. Bewegt man sich also mit seinem mobilen Gerät, so kann man zwar nicht darauf vertrauen, dass die Abdeckung jeder einzelnen Schnittstelle vollständig gewährleistet ist, jedoch besteht häufig die Möglichkeit, mit einer anderen Schnittstelle Verbindung zu bekommen. Als Beispiel sei hier ein Bahnhof genannt, bei dem man im Zug womöglich einen Bluetooth-Zugangspunkt nutzen kann, wohingegen am Bahnsteig und im Bahnhofsgebäude WLAN vorhanden ist. Verlässt man den Bahnhof muss man vermutlich auf ein Mobilfunknetz zurückgreifen, bis man in der Innenstadt erneut eine andere Netzwerktechnologie zur Verfügung hat, welche zwar einen kleineren Einflussbereich, dafür aber auch größere Datenrate und geringere Kosten hat. Eine vertikale Übergabe kann dann stattfinden, wenn sich ein mobiles Gerät im Einflussbereich mehrerer Zugangspunkte unterschiedlicher Funkzugangstechnologie befindet. Auch vor einer vertikalen Übergabe muss zunächst die Verbindung mit dem Ziel-

netzwerk stattgefunden haben. Wiederum erhält man eine Care-Of-Adresse, mit der dann ein Binding Update durchgeführt wird.

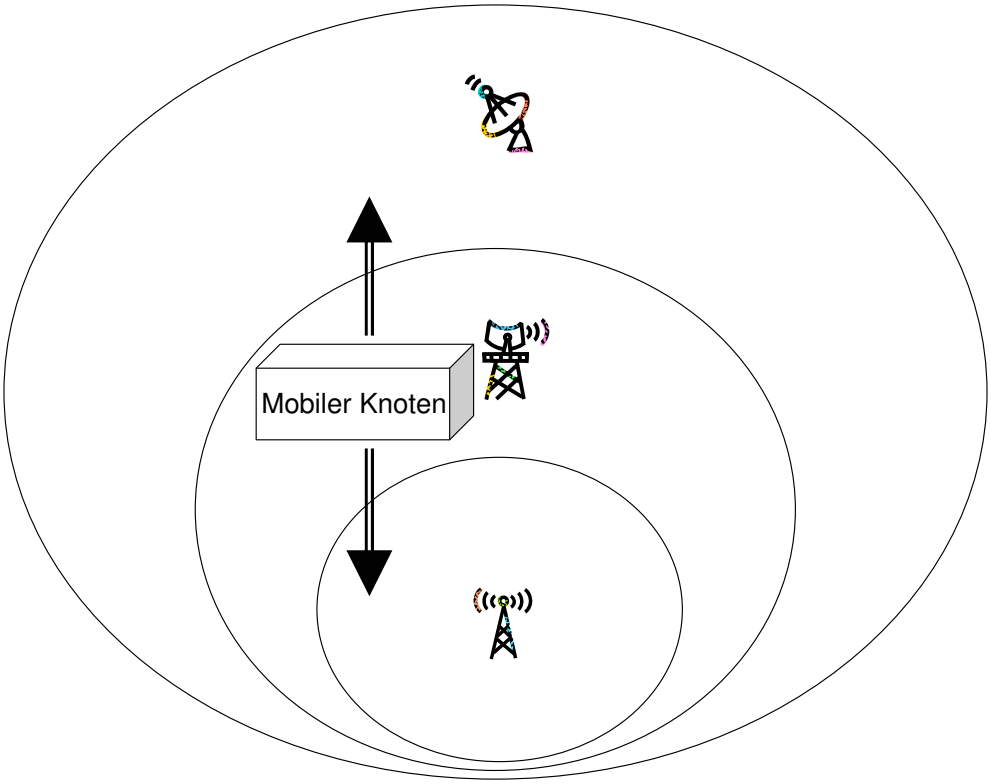

Abbildung 3: Abdeckungsbereich mehrerer Zugangspunkte unterschiedlicher Technologie mit vertikaler Übergabe

Für vertikale Übergaben gibt es mehr mögliche Szenarien, als bei horizontalen Übergaben. Des Weiteren muss man hierbei auch immer betrachten, in welche Richtung die Übergabe stattfindet. Eine Übergabe aufwärts bedeutet, zu einer größeren Zellgröße (WLAN->GPRS) hin, eine Übergabe abwärts bedeutet, zu einer kleinere Zellgröße (GPRS->WLAN) hin (siehe auch [StKa98] und Abbildung 3). Vertikale Übergaben aufwärts passieren meist, wenn die Verbindung der aktuellen Schnittstelle schwächer wird, oder zusammenbricht. Abwärts gerichtete Übergaben hingegen passieren, wenn Netze verfügbar werden. Verschiedene Faktoren steigern die Komplexität vertikaler Übergaben. Sind mehrere Netze mit ähnlichen Leistungsdaten verfügbar, muss eine Auswahl stattfinden, welcher Flow welche Schnittstelle nutzen soll. Ein weitere Faktor ist die angepeilte Granularität des Flow Movement. *Per-Correspondent-Node-Mobilität* ist die gröbste Granularitätsstufe und bedeutet, dass die gesamte Kommunikation mit einem Kommunikationspartner über eine Care-Of-Adresse läuft, auch wenn mehrere Datenverbindungen bestehen. So muss beim Kommunikationspartner pro mobilem Knoten nur ein Binding im Binding Cache stehen. *Per-Flow-Mobilität* ist die nächst feinere Stufe. Bei dieser kann der Kommunikationspartner für jeden Flow über eine andere Care-Of-Adresse mit demselben mobilen Knoten kommunizieren. Hierfür ist es notwendig, dass im Binding Cache für jeden Flow ein Binding gehalten wird. Die feinste bisherige Stufe nennt sich *Load-Balancing-Mobilität*. Dabei ist es möglich, einzelne Flows über mehrere Care-Of-Adressen an den mobilen Knoten zu senden. Im Binding Cache müssen hierzu für jeden Flow mehrere Bindings gehalten werden. Allerdings bringt diese Granularitätsstufe Probleme mit sich, die

bisher noch nicht zufriedenstellend gelöst werden konnten. Einerseits müsste der Datenstrom auf der Senderseite geeignet aufgeteilt werden, andererseits kann es durch unterschiedliche Laufzeiten auf den verschiedenen Schnittstellen zu Staus kommen, was die Zusammenführung des aufgeteilten Flows auf der Empfängerseite sehr schwierig macht. All diese Maßnahmen auf Schicht 3 durchzuführen ist sehr komplex und noch nicht ausreichend erforscht. Deshalb ist diese Granularitässtufe in der Aufzählung nur der Vollständigkeit wegen genannt und wird nicht weiter betrachtet.

4 Umsetzung von Flow Movement in Mobile IPv6 und die Limitierungen

Um Flow Movement umsetzen zu können, sind eine Reihe von Voraussetzungen im Protokoll nötig. Das aktuelle Protokoll ist IPv6. Dieses soll den immer noch am häufigsten verbreiteten Vorgänger IPv4 ablösen und bringt eine Menge von Neuerungen mit sich, vor allem die für diese Arbeit wichtige Erweiterung für mobile Geräte: Mobile IPv6 (siehe [ZSBK$^+$05]). Die wichtigsten Begriffe und Grundlagen, wie Heimatagent, Kommunikationspartner, mobiler Knoten und Bindings, sowie Binding Updates wurde in Abschnitt 2 bereits vorgestellt. Nun sollen diese Funktionsweisen mit den Voraussetzungen für Flow Movement zusammengeführt werden, um festzustellen, welche Möglichkeiten und welche Limitierungen es gibt.

Geht man von der Definition des Flow Movement und dem Beispiel in Abschnitt 1 aus, so lassen sich einige Anforderungen ableiten, die realisiert werden müssen, beziehungsweise sollten:

- Multihoming
- transparente Übergaben
- möglichst unterbrechungsfreie Übergaben
- Möglichkeit zur Einstellung von Präferenzen
- Granularität auf der Stufe von Per-Flow-Mobilität

Bei all diesen gilt es zu prüfen, ob und wie sie in Mobile IPv6 umgesetzt werden können.

Multihoming stellt hierbei schon ein erstes Problem dar, denn die momentane Implementierung von Mobile IPv6 unterstützt Multihoming nicht in vollem Umfang. In [MWEN$^+$05] wird detailliert auf diese fehlende Unterstützung eingegangen und es wurde versucht, einen kompletten Überblick über die Problemstellung zu geben. Aktuell wird daran gearbeitet geeignete Erweiterungen für Mobile IPv6 zu erstellen, welche diese Probleme beheben und Multihoming in möglichst vollständigem Umfang erlaubt. Eines der Probleme ist, dass zwar technisch gesehen, horizontale wie vertikale Übergaben in Mobile IPv6 unterstützt werden, aber vertikale Übergaben nicht vom Standard vorgesehen sind. Da aber auf der Ebene von Mobile IPv6 kein Unterschied zwischen horizontalen oder vertikalen Übergaben existiert, da sich schließlich nur die Care-Of-Adresse ändert, lassen sich vertikale Übergaben ohne weiteren Aufwand umsetzen.

Transparente Übergaben sind wichtig, damit Datenverbindungen keinen Abbruch erfahren. Applikationen können eine Änderung in der IP-Adresse nicht verarbeiten, daher ist es wichtig, dass sich oberhalb von Schicht 3, die Adressen nicht ändern. Mobile IPv6 leistet genau das. Damit ist die Anforderung nach Transparenz erfüllt.

Möglichst unterbrechungsfreie Übergaben bedürfen genauerer Betrachtung. Bei horizontalen Übergaben kann Mobile IPv6 durch Algorithmen zur Bewegungserkennung feststellen, ob ein mobiler Knoten sich von einem Subnetz in ein anderes bewegt hat. Diese Bewegungserkennung ist jedoch reaktiv und es kommt zu einer Unterbrechung des Datenstroms, bis die neue Verbindung aufgebaut ist. Um diese Unterbrechung zu umgehen, wären Änderungen am Mobile-IPv6-Protokoll nötig, welche die Bewegung des mobilen Knotens vorhersehen und eine neue Verbindung aufbauen, bevor die alte zusammenbricht. Bei vertikalen Übergaben kann die Bewegungserkennung nicht genutzt werden. Hier gibt es zwei Szenarien:

- Der mobile Knoten sieht mindestens zwei Netze und entscheidet, ohne äußere Einwirkung, dass eine Übergabe stattfinden soll (häufig bei abwärts gerichteten Übergaben). In diesem Fall muss die alte Verbindung so lange weiterbestehen, bis die neue Verbindung bereit ist, um Unterbrechungsfreiheit zu gewährleisten.
- Der mobile Knoten sieht mindestens zwei Netze und es kommt zum Verbindungsabbruch mit dem aktuellen Netz eines Flows. Selbst wenn der mobile Knoten auch mit den anderen Netzen bereits verbunden ist, so muss er erst ein Binding Update über die neue Schnittstelle senden, damit der Flow auf diesen weitergeleitet werden kann. Dies führt mindestens zu einer Unterbrechung im Datenfluss, im schlimmsten Fall aber zu einem Abbruch der Datenverbindung bei einem Timeout. Unterbrechungsfreiheit ist hier also nicht gegeben, bis hin zu einem völligen Vebindungsabbruch. Diese Limitierung besteht, da bei Mobile IPv6 im Binding Cache, pro mobilem Knoten, nur eine Care-Of-Adresse stehen kann und keine Möglichkeit besteht, eine weitere Adresse einzutragen oder zu löschen, welche benutzt wird, wenn Fehler bei der ersten Adresse auftreten. Für Erweiterungen von Mobile IPv6 besteht hier auf jeden Fall Handlungsbedarf.

In den vorhergehenden Abschnitten wurde häufiger von Präferenzen und Entscheidungen gesprochen, die bei der Wahl der Care-Of-Adresse für einen spezifischen Flow getroffen werden müssen. Nun muss zunächst geklärt werden, wie solche Präferenzen aussehen und wer die Entscheidung trifft. Die angesprochenen Präferenzen beziehen sich auf die Anforderungen von Flows, in Verbindung mit den Leistungsdaten von Netzwerktypen, beziehungsweise deren Schnittstellen. Der VoIP-Anruf aus dem Beispiel sollte möglichst nicht unterbrochen werden, braucht aber nur geringe Bandbreite. Deshalb eignet sich GPRS für solche Flows. Der Datendownload dagegen sollte möglichst viel Bandbreite bekommen, jedoch sind kurze Unterbrechungen, solange sie transparent sind, nicht problematisch. WLAN und Bluetooth sind deshalb geeignet. Diese Präferenzen müssen sowohl bei dem Initiator (mobiler Knoten) sowie bei der datenliefernden Instanz (je nach Routing der Heimatagent oder der Kommunikationspartner) vorhanden sein. Um dies zu erläutern, betrachte man sich zunächst den mobilen Knoten, der sich durch die Netze bewegt. Kommt ein neuer Zugangspunkt in Sicht, so muss der mobile Knoten wissen, ob eine der bestehenden Datenverbindungen bevorzugt über die neue Schnittstelle geleitet werden soll. Kommt es zu einer Unterbrechung mit einem bestehenden Zugangspunkt, so muss der mobile Knoten wissen, über welche der verbleibenden Schnittstellen der Datenfluss bevorzugt geleitet werden soll. Gleiches gilt für die Erstellung einer völlig neuen Datenverbindung. Auf der Seite des Heimatagenten oder des Kommunikationspartners, ist nur der Fall einer fehlerhaften Datenverbindung entscheidend. Dennoch wird auch hierfür die Information benötigt, über welche andere Care-Of-Adresse, der unterbrochene Datenstrom bevorzugt gesendet werden soll. Solche Präferenzen lassen sich als Filter darstellen, jedoch fehlt in Mobile IPv6 die Möglichkeit, solche an Bindings zu koppeln und zu versenden.

Die drei Granularitätsstufen werden von Mobile IPv6 nur teilweise unterstützt. Der normale Fall ist Per-Correspondent-Node-Mobilität, bei dem im Binding Cache pro Datenverbindung zwischen mobilem Knoten und Kommunikationspartner ein Binding registriert ist. So ist es

in der Spezifikation von Mobile IPv6 vorgesehen. Für Per-Flow-Mobilität ist es bereits nötig, für jeden Flow eine eigene Heimatadresse zu registrieren, um innerhalb der Spezifikation zu bleiben. Effektiver wäre die Möglichkeit, für jeden Flow ein eigenes Binding zu registrieren. Load-Balancing-Mobilität ist, unabhängig von der technischen Komplexität, mit der Spezifikation von Mobile IPv6 nicht möglich.

Diese Limitierungen machen deutlich, dass Mobile IPv6, so wie es bisher implementiert ist, nicht die Anforderungen erfüllt, um Flow Movement in vollem Umfang umzusetzen. Der folgende Abschnitt wird deshalb einige Ansätze zur Erweiterung von Mobile IPv6 vorstellen und erläutern, welche die angesprochenen Probleme und Limitierungen beheben sollen.

5 Vorgeschlagene Erweiterungen von Mobile IPv6 für Flow Movement

Die nun folgenden Erweiterungen von Mobile IPv6 zielen darauf ab, Flow Movement, oder zumindest einen gewissen Teil davon, im Netzwerkprotokoll Mobile IPv6 zu ermöglichen. Vier unterschiedliche Ansätze sollen hier kurz vorgestellt werden. Jeder wird in einem eigenen Abschnitt behandelt.

5.1 Mobile IPv6 für mehrere Schnittstellen (MMI)

Vollständige Unterstützung von Multihoming wurde als eine der wichtigsten Limitierungen bei der Realisierung von Flow Movement in Mobile IPv6 identifiziert. Besonders die Problematik mehrerer Schnittstellen und die nicht vorgesehenen vertikalen Übergaben sind Gründe dafür. In [MoNKL05] wird eine Erweiterung vorgestellt, die diese Problematik beheben soll. Mobile IPv6 für mehrere Schnittstellen, oder kurz MMI, beschreibt ein Szenario, bei dem ein mobiler Knoten mehrere unterschiedliche Schnittstellen zur Verfügung hat. Jedes von diesen hat eine globale IPv6-Adresse zugeordnet, über die der Knoten erreicht werden kann. Für ein solches Szenario werden Vorschläge gemacht, wie Flow Movement, insbesondere vertikale Übergaben, funktionieren können. Zunächst behandelt MMI jede dieser Schnittstellen mit ihren Verbindungen separat, so dass horizontale Übergaben über Subnetze wie üblich funktionieren. Vertikale Übergaben werden wie folgt realisiert: Ein mobiler Knoten hat zwei Schnittstellen I1 und I2. Diese sind über die IPv6-Adressen IP1 für I1 und IP2 für I2 in ihren Netzen angemeldet. Schnittstelle I1 ist verbunden mit Heimatagent HA1, Schnittstelle I2 ist verbunden mit Heimatagent HA2. Um die Übergabe eines Flows von I1 auf I2 durchzuführen, sendet die neue Schnittstelle I2 ein Binding Update an Heimatagent HA1 mit der alten IP1 als Heimatadresse und IP2 als neue Care-Of-Adresse im entsprechenden Feld. Bewegt sich dieser Knoten nun und es werden horizontale Übergaben nötig, so wird wie folgt verfahren: Bewegt sich I2 in ein neues Subnetz, so erhält sie eine neue IP3, welches als Binding Update an HA2 gesendet wird. Nach der Antwortprozedur wird dieses Binding Update an alle Kommunikationspartner vom mobilen Knoten weitergereicht, welche über die alte IP1 mit dem mobilen Knoten kommuniziert haben. Daher wird im Binding Cache von HA1 auch das neue Binding zwischen IP1 und IP3 registriert (siehe Abbildung 4).

Diese Mechanismen sind alle bereits in Mobile IPv6 enthalten und benötigen keine Erweiterungen, jedoch fehlt noch der Auslöser, um die vertikale Übergabe überhaupt einzuleiten. Hierfür wird vorgeschlagen, dass der mobile Knoten Tabellen verwaltet, mit denen er Flows und deren bevorzugte Schnittstellen assoziiern kann. Diese sind zusätzlich mit Prioritäten versehen, um bei Konflikten eine Entscheidung treffen zu können.

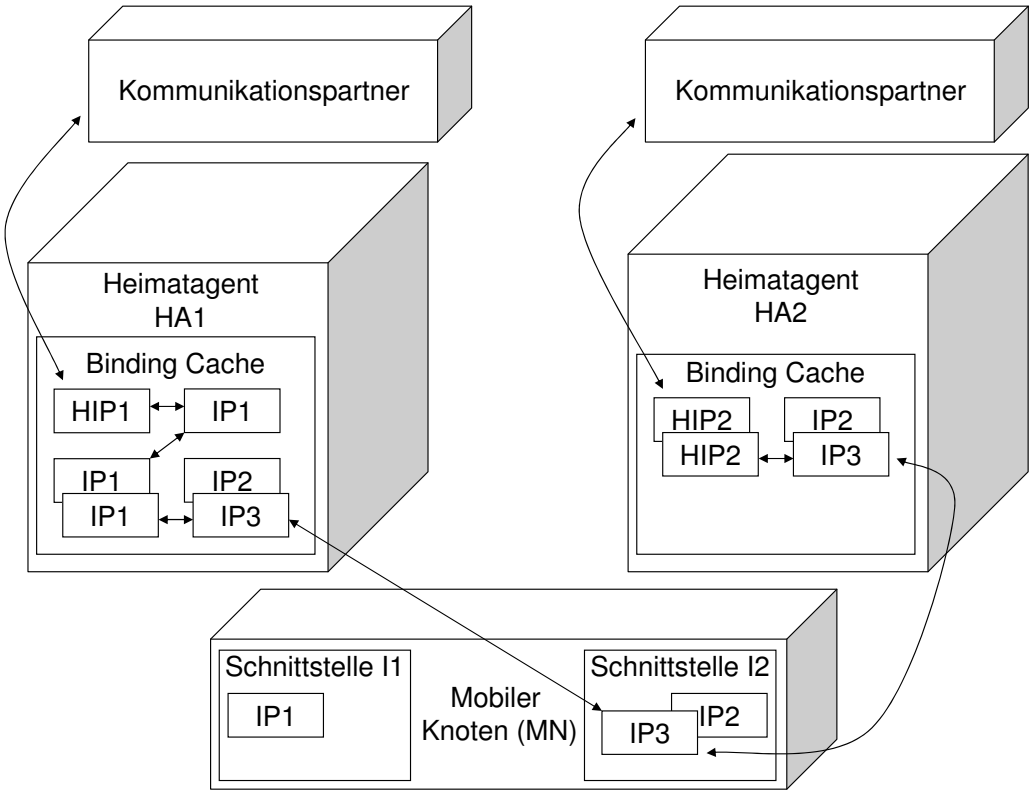

Abbildung 4: Datenfluss nach vertikaler und horizontaler Übergabe in der MMI Erweiterung

5.2 Mehrfache CoA-Registrierung

Mehrere Schnittstellen in einem mobilen Knoten lassen eine Menge theoretischer Vorteile erkennen, jedoch fehlen in Mobile IPv6 einige entscheidende Unterstützungen, um diese Vorteile praktisch umzusetzen. Eine davon ist die Einschränkung, jeder Heimatadresse nur eine Care-Of-Adresse zuordnen zu können. Deshalb wird in [WUEN05] eine Erweiterung für Mobile IPv6 vorgestellt, mit welcher es möglich wird, mehrere Care-Of-Adressen pro Heimatadresse zu registrieren. Hierzu sind einige Änderungen nötig, denn es gibt zwar bereits Unterstützung für das Vorhandensein von mehreren Heimatadressen pro mobilem Knoten, sowie das Vorhandensein von mehreren Care-Of-Adressen pro Heimatadresse, jedoch wird bei jedem Binding Update, der vorherige Eintrag im Binding Cache überschrieben. Zunächst muss also eine Möglichkeit geschaffen werden, um Bindings voneinander zu unterscheiden. Dies gelingt mit der Einführung einer neuen Binding ID (kurz BID). Diese muss im Binding Cache mit jedem Binding gespeichert werden, um die Identifizierung zu ermöglichen. Weiterhin muss im Binding Update eine Möglichkeit geschaffen werden, diese BID mitzusenden. Außerdem ist ein Schalter nötig, welcher angibt, ob das Binding mit dieser BID das primäre Binding sein soll, oder nicht. Da dies eine Erweiterung des Mobile-IPv6-Standards ist, kann man nicht davon ausgehen, dass jeder Kommunikationspartner diese Erweiterung unterstützt. Deshalb wird im Binding-Update-Mobility-Options-Feld ein zusätzlicher Schalter benötigt, welcher gesetzt wird, wenn die BID-Information im Binding Update enthalten ist. Kommunikationspartner, welche die Erweiterung beherrschen, können so die BID-Information auswerten, alle anderen Kommunikationspartner ignorieren sie einfach. Um zusätzliche Care-Of-Adressen zu registrieren, sendet der mobile Knoten ein Binding Update mit dem gesetzten BID-Schalter,

aber ohne BID, da diese vom Kommunikationspartner verwaltet wird. Unterstützt dieser die Erweiterung, so sendet er seine Antwort mit gesetzter BID. Der mobile Knoten muss diese dann in seinem Binding Cache registrieren. Um ein Binding Update eines speziellen Bindings durchzuführen, sendet der mobile Knoten das Binding Update wiederum mit gesetztem BID-Schalter, jedoch mit der entsprechenden BID. Der Kommunikationspartner findet den richtigen Eintrag in seinem Binding Cache über die BID und führt die Aktualisierung durch. Anschließend bestätigt er wie bisher. Einzelne Bindings können auch gelöscht werden, indem der mobile Knoten ein Binding Update mit der entsprechenden BID und einer Lebenszeit von null sendet. Um mehrfache CoA-Unterstützung zu beenden, sendet der mobile Knoten einfach ein normales Binding Update, ohne BID-Schalter, somit werden die Einträge im Binding Cache des Kommunikationspartners einfach überschrieben.

Zu diesen einfachen Änderungen, die nach außen sichtbar sind, gibt es natürlich noch einige interne Änderungen, welche aber transparent bleiben können. Als Ergebnis erhält man ein weitaus flexibleres Flow Movement, bei dem der Kommunikationspartner, bei Bedarf, einfach eine alternative Care-Of-Adresse für einen Flow wählen kann, ohne dass es zu Verbindungsabbrüchen kommt, und selbst die Unterbrechung kann auf ein Minimum reduziert werden. Zusätzlich zu diesem Vorschlag, beinhaltet die Arbeit einen weiteren Vorschlag, welcher die dynamische Heimatagent-Adress-Erkundung insofern modifiziert, dass dort auch ein BID-Schalter eingeführt wird, mit welchem der Heimatagent signalisieren kann, dass er die Erweiterung für mehrfache CoA-Registrierung unterstützt.

5.3 Flow-Movement-Optionen

Für Per-Flow-Mobilität, welche ja ein wichtiger Bestandteil einer vollständigen Unterstützung von Flow Movement ist, ist es notwendig, jedem Flow eine eigenes Binding zuzuordnen. In Mobile IPv6 ist dies nicht vorgesehen. Bei der Erweiterung für mehrfache CoA-Registrierung ist es zwar möglich, einer Datenverbindung zwischen mobilem Knoten und Kommunikationspartner mehrere alternative Bindings zu registrieren, aber dennoch gelten diese für alle Flows dieser Verbindung. Als grundlegende Anforderung für diese neue Problematik, erweist sich also die Möglichkeit, zur Unterscheidung einzelner Flows, damit diesen unterschiedliche Care-Of-Adressen zugewiesen werden können. In [SoEC03] wird eine Erweiterung mit dem Namen Flow-Movement-Optionen vorgestellt. Darin wird zunächst festgestellt, dass sich Flows auf zweierlei Weise einfach identifizieren lassen:

- einerseits über das Quintupel Zieladresse (der mobile Knoten), die Quelladresse, das Transportprotokoll, sowie die Port Nummern für Ziel und Quelle.
- andererseits über das Tupel Zieladresse (der mobile Knoten) und das in IPv6 definierte Flow Label.

Bei beiden Varianten ist die Zieladresse bereits im IP Header vorhanden, so dass bei der Erweiterung des Protokolls nur noch die anderen Informationen als Suboption im Anhang des Binding Updates gesendet werden muss, natürlich inklusive der üblichen Suboptionen, sowie einem Status für die Rückmeldung und der Adresse des Kommunikationspartners. Damit ist eine eindeutige Zuordnung von Bindings zu einzelnen Flows im Binding Cache möglich. Auch wenn nicht explizit angegeben, so ist wohl im Binding Update ein zusätzlicher Schalter nötig, mit dem signalisiert wird, dass Flow-Movement-Optionen enthalten sind, ähnlich wie in Abschnitt 5.2. Mit dieser Erweiterung lässt sich Per-Flow-Mobilität sehr viel effizienter umsetzen als es mit Mobile IPv6 bisher möglich war.

5.4 Filter-Registrierung mit NOMADv6

Mehrfach wurde bisher schon angesprochen, dass beim Flow Movement Entscheidungen getroffen werden müssen, welche Schnittstelle oder welche Care-Of-Adresse für welchen Flow oder welche Art von Flow genutzt werden soll. Diese Entscheidung kann nur auf Regeln basieren, die aufgestellt werden müssen. Ein sehr umfassender Ansatz führt das Prinzip dieser Regeln auf Filter zurück, welche an Bindings gekoppelt sind. Diese Erweiterung von Mobile IPv6 mit dem Namen NOMADv6 ist in [KuFG04] ausführlich beschrieben. Jedoch geht NOMADv6 sogar noch weiter und enthält auch die Möglichkeit mehrere Bindings pro mobilem Knoten zu halten, ähnlich wie in 5.2 beschrieben. Zu diesem Zweck wird ein neuer N-Schalter für das Binding Update vorgeschlagen, welches, wenn gesetzt, angibt, dass einerseits mehrere Bindings pro mobilem Knoten im Binding Cache gehalten werden müssen, und zum anderen, dass im Binding Update Filter als Suboption angeschlossen sind.

Bei einem Binding Update wird ja eine Heimatadresse mit einer Care-Of-Adresse assoziiert. Sind nun Filter-Regeln im Binding Update enthalten, so sind diese auch mit der entsprechenden Care-Of-Adresse zu assoziieren. Da mehrere Filter pro Binding Update enthalten sein können, werden die einzelnen Filter in Module aufgeteilt. Die Liste schließt dann mit einer speziellen Art von Filter-Modul ab. Die einzelnen Filter-Module sind per Definition als AND-verknüpft zu betrachten. Einzelne Filter-Module können mehrere Prädikate haben, welche wiederum OR-verknüpft sind. Die Filter-Erweiterungen, welche an ein Binding Update angehängt werden, können als vier unterschiedliche Typen kategorisiert werden:

- Filter-Modulerweiterung: Dies sind die Module, in denen die eigentliche Regeln enthalten sind. In NOMADv6 gibt es wiederum 10 verschiedene Arten. Dazu gehören Flow-Label-Filter, Protokoll-Filter, verschiedene Port-Filter und verschiedene Adress-Filter, um nur einige zu nennen. Diese Unterscheidung ist nötig, da sie jeweils unterschiedlich aussehen.
- Filter-Kontrollerweiterung: Dieses spezielle Modul enthält Kontrollinformation, die bei jedem Binding Update, welches Filter enthält, angehängt sein muss. Es erfüllt mehrere Aufgaben. Allen voran dient es als Abschluss der Filter-Modulliste im Binding Update. Zusätzlich enthält es wichtige Informationen wie den Filter-Index, zur Identifikation einzelner Filter im Binding Cache, sowie eine Gewichtung, mit welcher es möglich ist, einem Filter eine relative Menge von Netzwerkverkehr zuzuweisen.
- Filter-Löscherweiterung: Dieses spezielle Modul ist nötig um Filter bei Bedarf wieder aus dem Binding Cache zu entfernen.
- Filter-Bestätigungserweiterung: Mit diesem speziellen Modul wird das Ergebnis der Filter-Aktualisierung bei der Bestätigung des Binding Updates zurückgeliefert.

Diese Erweiterung bietet viele Möglichkeiten, die sich für Flow Movement nutzen lassen. Einiges an Funktionalität aus den anderen Erweiterungen lässt sich damit ebenfalls erreichen. Jedoch wächst auch die zusätzliche Datenmenge, die bei einem Binding Update gesendet werden muss.

Literatur

[EMWP+05] T. Ernst, N. Montavont, R. Wakikawa, E. Paik, C. Ng, K. Kuladinithi und T. Noel. Goals and Benefits of Multihoming. Internet-Draft, Februar 2005.

[KuFG04] K. Kuladinithi, N. A. Fikouras und C. Goerg. Filters for Mobile IPv6 Bindings (NOMADv6). Internet-Draft, Mai 2004.

[MoNKL05] N. Montavont, T. Noel und M. Kassi-Lahlou. Mobile IPv6 for multiple Interfaces (MMI). Internet-Draft, Juli 2005.

[MWEN+05] N. Montavont, R. Wakikawa, T. Ernst, C. Ng und K. Kuladinithi. Analysis of Multihoming in Mobile IPv6. Internet-Draft, Juni 2005.

[SoEC03] Hesham Soliman, Karim ElMalki und Claude Castelluccia. Flow Movement in Mobile IPv6. Internet-Draft, Juni 2003.

[StKa98] Mark Stemm und Randy H. Katz. Vertikal Handoffs in wireless overlay networks. Internet-Draft, 1998.

[WUEN05] Ryuji Wakikawa, Keisuke Uehara, Thierry Ernst und Kenichi Nagami. Multiple Care-of Adresses Registration. Internet-Draft, Juni 2005.

[ZSBK+05] Prof. Martina Zitterbart, Oliver Stanze, Peter Baumung, Tobias Küfner und Christian Vogt. Mobilkommunikation (Vorlesung), 2005.

Abbildungsverzeichnis

802.21 - Medienunabhängige Handover

Kevin Künzel

Kurzfassung

Diese Arbeit zum Seminar „Mobiles Internet“ befasst sich mit dem in [IEEE05] beschriebenen kommenden Standard 802.21 - Medienunabhängige Handover. Dieser Standard befindet sich zur Zeit noch in Arbeit und hat das Ziel, ein nahtloses Handover in homogenen und heterogenen Netzwerken zu ermöglichen. Dafür wird eine neue Zwischenschicht im ISO/OSI-Referenzmodell eingeführt, welche es auch in heterogenen Netzwerken ermöglicht, Informationen von den unteren Schichten bereitzustellen, die die Entscheidung für Handover erleichtern. Diese Arbeit beschreibt die neue Zwischenschicht näher und erklärt auch die Dienste und Primitiven, die diese Schicht den höheren Schichten zur Verfügung stellt. Auf das dazugehörige Protokoll samt Paketformat wird auch näher eingegangen.

1 Einleitung

Das IEEE (Institute of Electrical and Electronics Engineers) ist ein angesehener internationaler Verband von Ingenieuren. Es veranstaltet Fachtagungen, gibt Fachzeitschriften heraus und bildet Gremien zur Standardisierung von Technologien. Eines ihrer Projekte ist das IEEE 802, welches sich mit Standards aus dem Bereich der lokalen Netze (LANs und MANs) beschäftigt. Das Projekt ist wiederum in kleinere Arbeitsgruppen unterteilt, die sich regelmäßig auf Konferenzen treffen und vor allem auch über Konferenzschaltungen und per Email über ihr jeweiliges Thema diskutieren. Die Mitglieder dieser Arbeitsgruppen stammen hauptsächlich aus der Industrie und versuchen deshalb möglichst große Teile ihrer eigenen Entwicklungen in den abschließenden Standard hinein zu bekommen. Die einzelnen Arbeitsgruppen geben auch Änderungswünsche und Empfehlungen an andere Gruppen weiter.
Die Arbeitsgruppe 21 wurde im März 2004 gegründet und beschäftigt sich mit medienunabhängigen Handovern. Derzeitige Standards bieten nur Handover zwischen Zugangspunkten derselben Technologie an (z.B. innerhalb eines WLANs oder in einem Mobilfunknetz). In Zukunft soll es aber auch möglich sein zwischen Zugangspunkten unterschiedlicher Technologien zu wechseln. So könnte z.B. ein mobiles Endgerät ohne Abbruch einer aktiven Verbindung aus einem 3G-Mobilfunknetz in ein WLAN wechseln, wenn dieses in Reichweite kommt. 802.21 will solche Handover ermöglichen indem zwischen Schicht 2 und 3 des ISO/OSI-Referenzmodells ein Art Zwischenschicht eingefügt wird. Sie soll die Netzwerkerkennung und -auswahl erleichtern, die Sicherheit der Kommunikation soll gewährleistet bleiben und Energie soll gespart werden (z.B. indem Scanvorgänge nach verfügbaren Netzen reduziert werden). Der Standard 802.21 ist derzeit noch in Arbeit, allerdings wird erwartet, dass Standardisierung im März 2007 abgeschlossen sein wird.

2 Allgemeine Architektur

Um die Ziele aus Abschnitt 1 zu erreichen, stellt 802.21 eine in [IEEE05] beschriebene Architektur zur Verfügung, die es ermöglichen soll, den Dienst aufrecht zu erhalten, während das

mobile Endgerät zwischen verschiedenen Zugangspunkten heterogener Technologien wechselt. Um dies zu erreichen, wird zwischen Schicht 2 und 3 des ISO/OSI-Referenzmodells eine Zwischenschicht eingeführt, die den Netzwerken schon auf Schicht 2,5 mehr „Intelligenz“ bereitstellen soll. Diese Schicht wird medienunabhängige Handover-Funktion genannt und stellt den höheren Schichten neue Dienste zur Verfügung, um Handover in heterogenen Netzwerken zu ermöglichen und zu erleichtern.

2.1 MIH Funktion

Die medienunabhängige Handover-Funktion (MIH-Funktion, Media Independend Handover Function) ist zwischen der MAC-Schicht und der Vermittlungsschicht angesiedelt. Sie befindet sich sowohl auf dem mobilen Knoten als auch auf den Netzwerkelementen, die die Mobilität unterstützen (z.B. Access Point eines WLANs). Die MIH-Funktion bietet den höheren Schichten eine einheitliche Schnittstelle um auf die unterschiedlichen Informationen der zur Verfügung stehenden heterogenen Zugangstechnologien zugreifen zu können. Die Dienstprimitiven dieser Schnittstelle sind unabhängig von den Zugangstechnologien. Auf die Dienste der niedrigeren Schichten greift die MIH-Funktion über die bereits in anderen Standards definierten medienabhängigen Dienstzugangspunkte zu.

2.2 Dienste

Wie in [IEEE05] beschrieben, stellt die MIH-Funktion sowohl synchrone als auch asynchrone Dienste über wohldefinierte Dienstzugangspunkte für niedrigere und höhere Dienste zur Verfügung. Diese Dienste sollen den Protokollen der höheren Schichten (z.B. Mobile IP) helfen, dem Benutzer einen ununterbrochenen Dienst bereitzustellen, den Dienst an variierende Dienstgüten anzupassen, Energie zu sparen und die Netzwerkerkennung und -auswahl zu erleichtern. Die MIH-Funktion stellt folgende 3 Dienste bereit:

- Medienunabhängiger Ereignisdienst (Media Independent Event Service): Der medienunabhängige Ereignisdienst ist ein asynchroner Dienst. Er unterstützt sowohl lokale als auch entfernte Ereignisse sofern der entfernte Knoten denselben Medientyp hat. Ereignisse können Statusänderungen oder Änderungen des Übertragungsverhaltens der Schicht 2-Datenverbindung anzeigen. Typischerweise wird dieser Dienst verwendet, um die Netzwerkerkennung innerhalb eines Mobilitätsprotokolls zu erleichtern. Beispiele für Ereignisse sind „Link Up“, „Link Down“ oder auch „Link Going Down“ (Erklärungen zu den Ereignissen folgen in Abschnitt 3.1). Die Ereignisse werden von der Schicht 2-Datenverbindung (MAC-Schicht), der PHY-Schicht (Schicht 1) oder einer MIH-Funktion (auch von einer entfernten MIH-Funktion) ausgelöst. Das Ziel ist entweder die lokale MIH-Funktion, eine entfernte MIH-Funktion oder beides. Um über Ereignisse informiert zu werden, ist eine vorherige Registrierung notwendig.

- Medienunabhängiger Befehlsdienst (Media Independent Command Service): Der medienunabhängige Befehlsdienst ist ein synchroner Dienst. Er wird verwendet, um Befehle von höheren Schichten an niedrigere Schichten des Referenzmodells zu ermöglichen. Die Befehle können entweder von höheren Schichten an eine MIH-Funktion (z.B. von einem Mobilitätsprotokoll an eine MIH-Funktion) oder von der MIH-Funktion an niedrigere Schichten (z.B. an die MAC-Schicht) gerichtet sein.

- Medienunabhängiger Informationsdienst (Media Independent Information Service): Beim medienunabhängigen Informationsdienst handelt es sich um einen synchronen Dienst. Er stellt ein Rahmenwerk und dazugehörige Mechanismen bereit, um es der

MIH-Funktion zu ermöglichen, Netzwerkinformationen innerhalb eines geographischen Gebietes zu entdecken und zu erlangen. Dieser Dienst stellt in erster Linie eine Menge von Informationselementen, die Informationsstruktur und ihre Repräsentation sowie einen Anfrage/Antwort-Mechanismus zum Informationsaustausch zur Verfügung. Die Speicherung der Informationen soll entweder innerhalb der MIH-Funktion (also im Stack) oder auf einem externen Informationsserver erfolgen. Die Informationen sind sowohl den höheren als auch den niedrigeren Schichten zugänglich. Der Austausch der Informationen soll im TLV-Format (Typ-Länge-Wert-Format) erfolgen. Die Informationen dieses Dienstes beinhalten sowohl Informationen über MAC-Adressen und Sicherheit als auch über verfügbare Dienste höherer Schichten eines Netzwerks. Der Informationsdienst soll einen schnellen Informationsaustausch mit möglichst wenig Overhead ermöglichen. Außerdem ist ein Informationszugriff ohne große Berechnungen erwünscht.

2.3 Referenzmodelle

In diesem Abschnitt werden die Referenzmodelle für die Netzwerke der Standards 802.3 (Ethernet), 802.11 (WLAN), 802.16 (WIMAX) und für die 3G-Mobilfunknetze (GPRS/UMTS) kurz vorgestellt. Dabei werden die Änderungen, die in [IEEE06b] beschrieben sind, berücksichtigt. Die Referenzmodelle zeigen auf, wo sich die Dienstzugangspuntke und Schnittstellen in den unterschiedlichen Netzwerken befinden und wie die medienunabhängige Handover-Schicht eingefügt wurde.

2.3.1 Referenzmodell für 802.3

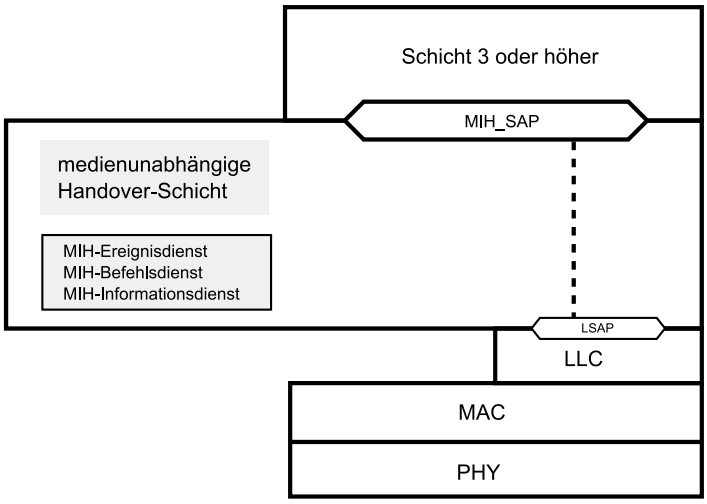

Abbildung 1: Medienunabhängiges Handover-Referenzmodell für 802.3

Abbildung 1 zeigt das Referenzmodell für das Ethernet (IEEE 802.3). Der Standard 802.21 unterstützt den medienunabhängigen Ereignisdienst, den medienunabhängigen Befehlsdienst und den medienunabhängigen Informationsdienst. Die Nutzdaten der MIH-Funktion werden über die Datenebene verschickt, indem bereits existierende Primitiven benutzt werden. Dafür wird der Dienstzugangspunkt LSAP genutzt. Über den Dienstzugangspunkt MIH_SAP ist es für die höheren Schichten möglich, auf die Dienste der MIH-Funktion zuzugreifen.

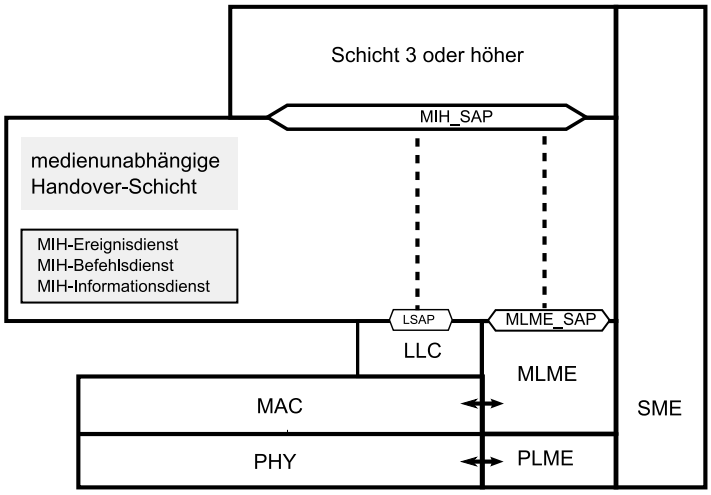

Abbildung 2: Medienunabhängiges Handover-Referenzmodell für 802.11

2.3.2 Referenzmodell für 802.11

Wie in Abbildung 2 zu sehen ist, definiert LSAP den Dienstzugangspunkt der MIH-Funktion zur Datenebene. Er kann MIH-Nutzdaten in Datenpaket kapseln. Da ein Versand von Nachrichten allerdings erst nach erfolgreicher Verbindung auf Schicht 2 möglich ist, wird vorgeschlagen, den Dienstzugangspunkt MLME_SAP mit zusätzlichen Handover-Fähigkeiten zu erweitern, um einen Transport über die Managementebene zu ermöglichen. Über den Dienstzugangspunkt MIH_SAP ist ein Zugriff der höheren Schichten auf die Dienste der MIH-Funktion möglich.

2.3.3 Referenzmodell für 802.16

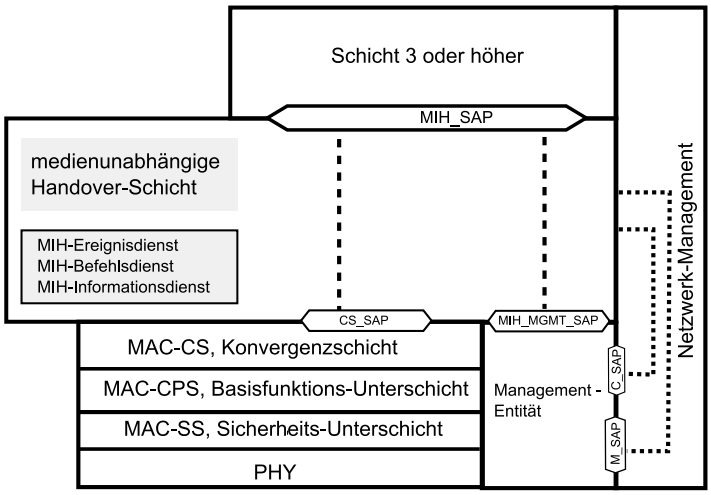

Abbildung 3: Medienunabhängiges Handover-Referenzmodell für 802.16

Abbildung 3 zeigt das Referenzmodell für Systeme, die auf dem Standard 802.16 (WIMAX) basieren. Die MIH-Funktion und das Netzwerkkontroll-Managementsystem haben die gemeinsamen Dienstzugangspunkte M_SAP und C_SAP. Über sie erfolgt die Kommunitation zwischen der MIH-Funktion und der Management-Entität und sie hilft bei dem Transport der

MIH-Nutzdaten. Über den Dienstzugangspunkt MIH_SAP erfolgt die Kommunikation mit den höheren Schichten.

2.3.4 Referenzmodell für 3G-Mobilfunknetze

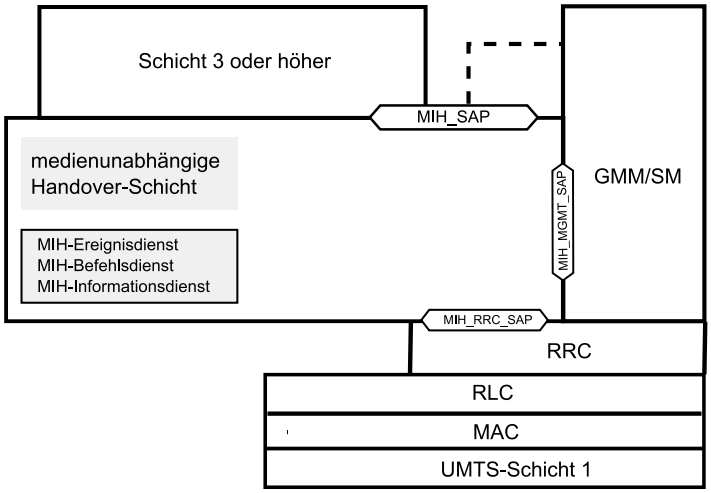

Abbildung 4: Medienunabhängiges Handover-Referenzmodell für 3G-Mobilfunknetze

In Abbildung 4 ist eine mögliche Realisation für ein 3G-Mobilfunknetz (GPRS/UMTS) dargestellt. Der Dienstzugangspunkt MIH_RRC_SAP definiert den Zugangspunkt zur RRC-Schicht (Radio Resource Control) der 3G-Netze. Über den Dienstzugangspunkt MIH_MGMT_SAP ist ein Zugriff auf das GPRS-Mobilitäts-Management (GMM) bzw. das Session-Management (SM) der 3G-Netze möglich. Die meisten MIH-Dienste werden die Informationen nutzen, die bereits durch die RRC-Schicht und das Mobilitäts- bzw. Session-Management definiert sind. Es ist möglich, dass die GMM/SM-Schicht in Zukunft erweitert wird, indem man die Funktionalität, die durch den MIH_MGMT_SAP-Dienstzugangspunkt zur Verfügung gestellt wird, nutzt.

3 Dienste der MIH-Funktion

Die MIH-Funktion stellt den medienunabhängigen Ereignisdienst, den medienunabhängigen Befehlsdienst und den medienunabhängigen Informationsdienst bereit. In [IEEE05] und den nächsten Abschnitten werden diese Dienste näher beschrieben.

3.1 Medienunabhängiger Ereignisdienst

Im Allgemeinen können Handover entweder von dem mobilen Endgerät oder von dem Netzwerk eingeleitet werden. Die Ereignisse, die ein Handover anstoßen, können also von der physikalischen Schicht, der MAC-Schicht oder der MIH-Funktion des mobilen Knotens oder des Netzzugangspunktes stammen. Gründe für ein Handover könnten die Mobilität des Nutzers bzw. des Endgerätes sein oder auch die Veränderung der Umgebungsbedingungen. Es ist möglich, dass die Ereignisse an mehrere Ziele und unterschiedliche Schichten geliefert werden sollen. Die MIH-Funktion kann bei dieser Verteilung helfen. Dafür können sich höhere Schichten bei der MIH-Funktion registrieren, um über Ereignisse bestimmter Quellen informiert zu werden. Bei den Ereignissen handelt es sich um diskrete Ereignisse, d.h. es gibt im

Allgemeinen keinen Ereignisstatus. Allerdings können hierfür Identifikatoren benutzt werden und andere Ereignisse können über diese Identifikatoren verbunden werden. Aus der Sicht des Empfängers sind die Ereignisse nur beratender Natur, d.h. es besteht kein Zwang, auf Ereignisse zu reagieren. Außerdem muss auf höheren Schichten die Zuverlässigkeit und Robustheit der Ereignisse überprüft werden. So könnten höhere Protokolle „defensiver“ auf Ereignisse reagieren, wenn diese von entfernten Entitäten stammen. Es gibt mehrere Ereignistypen (siehe auch [IEEE05]):

- MAC- und PHY-Statusänderungs-Ereignisse: Diese Ereignisse entsprechen den Veränderungen des Status der MAC- bzw. der physikalischen Schicht. Ein Beispiel dafür ist das „Link Up“-Ereignis (Verbindung hergestellt).

- Vorhersagende Ereignisse (Predictive Events): Vorhersagende Ereignisse drücken ausgehend von vergangenen oder derzeitigen Bedingungen die Wahrscheinlichkeit einer Veränderung in der Zukunft aus. So könnte z.B. eine Verschlechterung der Signalstärke eines WLAN-Netzwerks den baldigen Verbindungsabbruch zu diesem Netzwerk bedeuten. Da diese Ereignisse nur versuchen, die Zukunft vorauszusagen, ist es natürlich möglich, dass die Vorhersagen unzutreffend sind. Deshalb können sie Zeitgrenzen, in denen das Ereignis eintreffen soll, und einen Zuverlässigkeitsgrad, der die Wahrscheinlichkeit des Eintreffens des vorausgesagten Ereignisses angibt, beinhalten.

- Lokale und entfernte Ereignisse (Local/Remote Events): Lokale Ereignisse haben ihren Ursprung im lokalen Speicher. Entfernte Ereignisse kommen von MIH-Funktionen ausserhalb des lokalen Speichers.

Um über Ereignisse informiert zu werden, muss vorher eine Registrierung für dieses Ereignis stattgefunden haben.

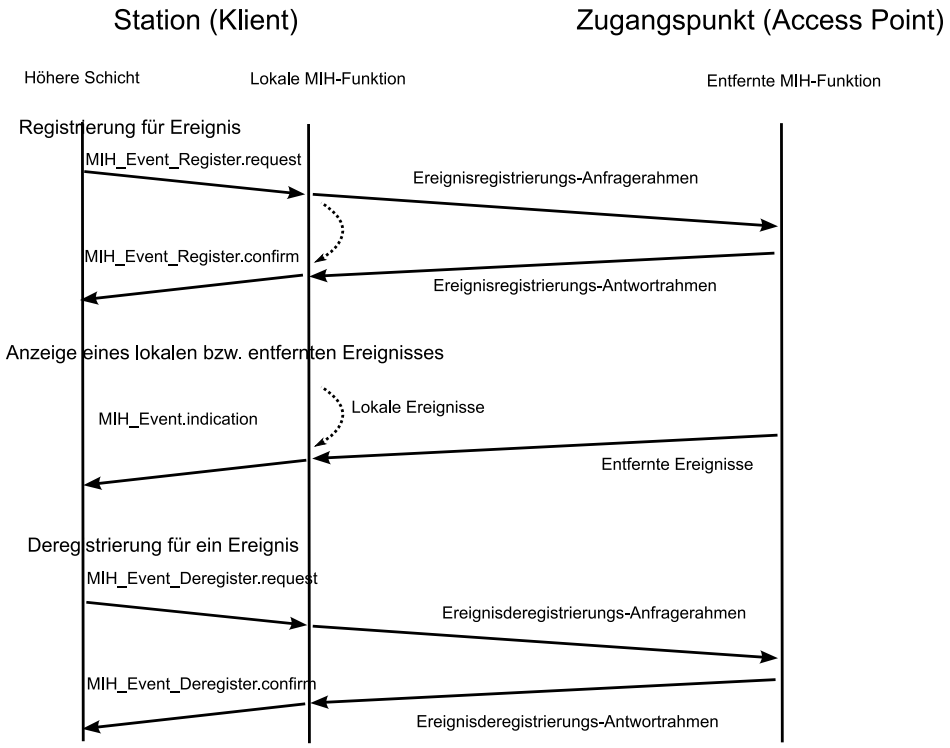

Abbildung 5: Flussdiagramm für MIH-Ereignisse

In Abbildung 5 ist das Flussdiagramm für MIH-Ereignisse zu sehen. Bei der Registrierung für ein Ereignis bzw. der Deregistrierung von einem Ereignis wird eine Bestätigung erwartet. Die Ereignisse selbst werden nur angezeigt (Indication) und erwarten keine Antwort. Derzeit sind 11 Ereignisse vorgesehen:

- Link Up und Link Down: Die Verbindung zu einem Netzwerk wurde hergestellt bzw. unterbrochen.
- Link Going Down: Die Verbindung wird wahrscheinlich bald unterbrochen (vorhersagendes Ereignis).
- Link Detected und Link Parameters Change: Eine neue nutzbare Verbindung wurde entdeckt bzw. die Verbindungsparameter (z.B. Geschwindigkeit, Dienstgüte) haben sich geändert.
- Link Event Rollback: Wird zusammen mit „Link Going Down“ benutzt. Falls die Verbindung doch nicht abbricht, werden die veranlassten Änderungen rückgängig gemacht oder ignoriert.
- Link SDU Transmit Success und Link SDU Transmit Failure: Falls Pakete höherer Schichten in mehrere Pakete aufgeteilt werden mussten, zeigen diese Ereignisse an, dass das gesamte Paket übertragen wurde bzw. die Übertragung fehlgeschlagen ist.
- Link Handoff Imminent, Link Handoff Proceeding und Link Handoff Complete: Wird generiert, bevor das Handover stattfindet bzw. während oder nach dem Handover. Diese Ereignisse beinhalten unter anderem Informationen über den neuen Netzzugangspunkt.

3.2 Medienunabhängiger Befehlsdienst

Der medienunabhängige Befehlsdienst bezieht sich auf Befehle, die von höheren Schichten an niedrigere Schichten des Referenzmodells gesendet werden. Der Dienst kann, wie in [IEEE05] zu sehen ist, verwendet werden, um den Status von Verbindungen zu bestimmen und eine optimale Leistung zu erreichen. Die Informationen die von dem Dienst bereitgestellt werden, sind vor allem dynamische Informationen wie Verbindungsparameter (z.B. Signalstärke, Verbindungsgeschwindigkeit, usw.). Im Gegensatz dazu stellt der medienunabhängige Informationsdienst hauptsächlich eher statische Informationen wie Netzwerkoperator oder verfügbare Dienste höherer Schichten bereit. Die Informationen des medienunabhängigen Befehlsdienstes und des Informationsdienstes zusammen können benutzt werden, um das Handover zu erleichtern. Es wurden einige Befehle hinzugefügt, um es den höheren Schichten zu ermöglichen, niedrigere Schichten zu konfigurieren, zu kontrollieren und Informationen zu erhalten. Es gibt zwei Arten von Befehlen:

- MIH-Befehle: Diese Befehle werden von den höheren Schichten an die MIH-Funktion gegeben. Dabei kann sowohl die lokale als auch eine entfernte MIH-Funktion das Ziel sein.
- Verbindungsbefehle: Diese Befehle haben die niedrigeren Schichten als Ziel und werden von der MIH-Funktion abgesetzt. Auch hier können sowohl lokale als auch entfernte niedrigere Schichten das Ziel sein.

Abbildung 6 zeigt das Flussdiagramm für MIH-Befehle. Man sieht, dass jeder Befehl bestätigt werden muss. Es gibt folgende MIH-Befehle:

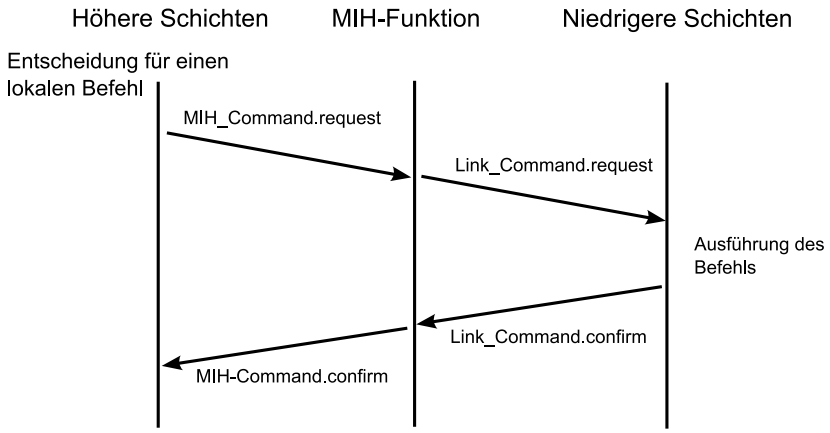

Abbildung 6: Flussdiagramm für MIH-Befehle

- MIH Poll: Dieser Befehl wird benutzt, um den Status von derzeit benutzten und potentiell verfügbarer Verbindungen abzufragen.
- MIH Switch: Mit diesem Befehl kann zwischen zwei verschiedenen Verbindungen gewechselt werden.
- MIH Configure: Dieser Befehl wird benutzt, um das Verhalten von Verbindungen zu konfigurieren.
- MIH Scan: Mit dem Scan-Befehl wird nach benachbarten Netzzugangspunkten gesucht, die evtl. später benutzt werden können.
- MIH Handover Initiate: Mit diesem Befehl kann eine MIH-Funktion ihre Handover-Absicht an eine entfernte MIH-Funktion bekannt geben.
- MIH Handover Prepare: Dieser Befehl wird von der MIH-Funktion des alten Netzzugangspunktes benutzt um Resourcen auf dem neuen Netzzugangspunkt anzufordern und ihn auf das Handover vorzubereiten.
- MIH Handover Commit: Mit diesem Befehl wird die Handover-Absicht bestätigt.
- MIH Handover Complete: Dieser Befehl wird von dem neuen Netzzugangspunkt genutzt, um der MIH-Funktion des alten Netzzugangpunktes mitzuteilen, dass das Handover abgeschlossen ist.
- MIH Handover Network Address Information: Dieser Befehl kann von der MIH-Funktion des alten Netzzugangpunktes genutzt werden, um Informationen über die Netzwerkadresse des mobilen Endgeräts vor dem Handover an den neuen Netzzugangspunkt zu senden.

3.3 Medienunabhängiger Informationsdienst

Der medienunabhängige Informationsdienst stellt, wie in [IEEE05] beschrieben ist, ein Rahmenwerk bereit, um es der MIH-Funktion sowohl im mobilen Knoten als auch im Netzwerk zu ermöglichen, Informationen über homogene oder heterogene Netzwerke innerhalb eines Gebietes zu entdecken und zu erhalten. Mit diesen Informationen soll dann das Handover in diesen Netzwerken erleichtert werden. Das Ziel ist es, eine globale Sicht über die heterogenen Netzwerke zu erlangen um ein nahtloses Handover in diesen Netzwerken zu erleichtern. Es gibt zwei Mobilitätsarten:

- Horizontale Handover: Ein horizontales Handover wird erreicht, wenn zwischen verschiedenen Netzzugangspunkten derselben Funktechnologie gewechselt wird (z.B. zwischen verschieden Access Points bei 802.11).
- Vertikale Handover: Bei einem vertikalen Handover wird von einem Netzwerktyp zu einem anderen Netzwerktyp gewechselt, z.B. von WLAN zu GPRS.

Der medienunabhängige Informationsdienst beinhaltet die Unterstützung von verschiedenen Informationselementen. Diese Informationselemente stellen Informationen bereit, die benötigt werden, um intelligente Handoverentscheidungen zu treffen. Abhängig von der Art des Handovers sind verschiedene Informationen notwendig. Bei horizontalen Handovern zwischen verschiedenen Netzzugangspunkten desselben Netzwerktyps sind Informationen, die von den niedrigeren Schichten des Zugangsnetzwerks bereitgestellt werden, meistens ausreichend, da die angebotenen Dienste höherer Schichten sich dabei nicht ändern. Bei vertikalen Handovern bewegt sich das Endgerät jedoch zwischen verschiedenen Zugangspunkten unterschiedlicher Technologien. Dabei ist es nötig bei der Auswahl eines neuen Netzzugangspunktes darauf zu achten, dass Dienste höherer Schichten angeboten werden, die von derzeit aktiven Anwendungen benötigt werden. Der medienunabhängige Informationsdienst ist imstande, solche Informationen zu erlangen. Dies beinhaltet sowohl Informationen niedrigerer Schichten (z.B. Verbindungsparameter) als auch Informationen über verfügbare Dienste höherer Schichten (z.B. Verfügbarkeit von VPN-Diensten oder bestimmter Verschlüsselungsverfahren). Der Informationsdienst sammelt Informationen über verschiedene Netzwerke und stellt diese Informationen jedem einzelnen Netzwerk zur Verfügung. Dadurch werden energie- und zeitvergeudende Scanvorgänge vermieden, die ohne den Informationsdienst nötig gewesen wären. Es gibt drei Arten von Informationselementen:

- Allgemeine Netzwerkinformationen (General Network Information): Diese Informationselemente geben einen allgemeinen Überblick über das Netzwerk wie Netzwerk-ID, Standort von Netzzugangspunkten, IP-Version, usw.
- Information der Verbindungsschicht (Link Layer Information): Informationen der Verbindung, z.B. Verbindungsparameter wie Kanal oder Frequenz, Datenraten, Dienstgüte, usw.
- Informationen höherer Schichten: Informationen über angebotene Dienste höherer Schichten oder Applikationen, die unterstützt werden, z.B. Multimedia-Message-Service (MMS), Mobile IP, Voice-over-IP (VoIP), usw.

Auch wenn all diese Informationselemente benötigt werden, so wird der Zugriff auf Informationen der Verbindungsschicht meistens direkt über medienspezifische Technologien durchgeführt werden. Nur wenn dies nicht möglich ist, werden die entsprechenden Informationselemente benutzt. Die Informationen sollen im TLV-Format (Typ-Länge-Wert-Format) ausgetauscht werden.

In Abbildung 7 ist das Flussdiagramm für Informationen des medienunabhängigen Informationsdienstes zu sehen. Der Informationsdienst innerhalb der MIH-Funktion kommuniziert mit der entfernten MIH-Funktion z.B. eines Zugangspunkts (Access Points) oder einer Basisstation (Base Station).

4 Medienunabhängiges Handover-Protokoll

Die MIH-Funktion stellt synchrone und asynchrone Dienste bereit, die über wohldefinierte Dienstzugangspunkte für höhere Schichten erreichbar sind. Die Dienstzugangspunkte beinhal-

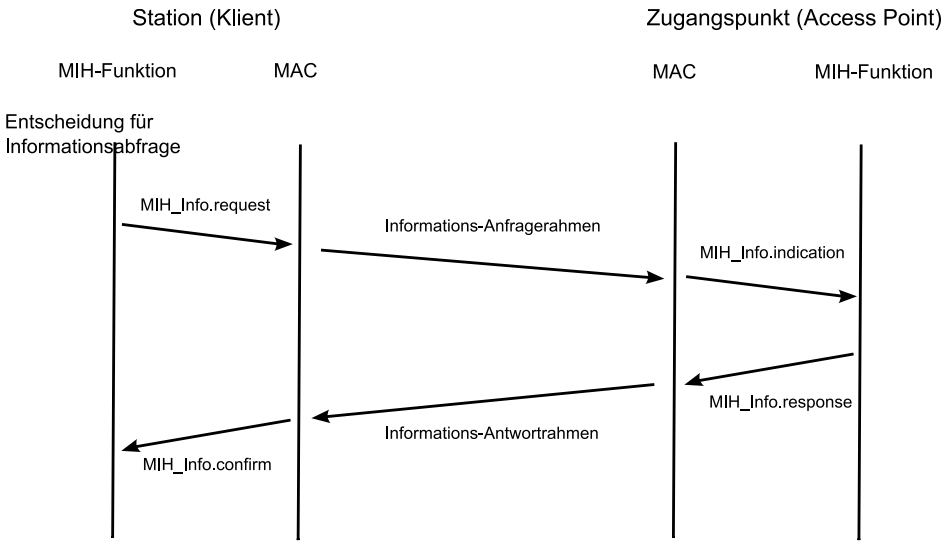

Abbildung 7: Flussdiagramm für MIH-Informationen

ten den Dienstzugangspunkt für höhere Schichten, über den die Nutzer des medienunabhängigen Handovers auf die Dienste der MIH-Funtion zugreifen können, und die Dienstzugangspunkte zu den niedrigeren Schichten, über die die MIH-Funktion auf die medienspezifischen Ressourcen zugreifen und sie kontrollieren kann. Das in [IEEE05] beschriebene medienunabhängige Handover-Protokoll definiert Rahmenformate mit denen Nachrichten zwischen benachbarten MIH-Funktionen ausgetauscht werden können. Diese Nachrichten basieren auf den in den Abschnitten 3.1, 3.2 und 3.3 beschriebenen Primitiven. Die Nachrichten zu bzw. von niedrigeren Schichten werden über Datenrahmen verschickt.

4.1 Dienste des medienunabhängigen Handover-Protokolls

Das medienunabhängige Handover-Protokoll stellt folgende Dienste bereit:

- Entdeckung von MIH-Fähigkeit: Die MIH-Funktion im mobilen Endgerät oder im Netzwerk kann entdecken, welche Entitäten medienunabhängige Handover unterstützen. Danach können die benachbarten MIH-Funktionen optimale Verbindungsparameter/-wege aushandeln. Außerdem können Listen über unterstützte Ereignisse und Befehle sowie Informationselemente ausgetauscht werden.

- Entfernte MIH-Registrierung: Entfernte MIH-Funktionen in unterschiedlichen Entitäten können sich untereinander registrieren, um medienunabhängige Nachrichten zu erhalten. Für den medienunabhängigen Befehlsdienst ist keine Registrierung nötig.

- Austausch von MIH-Nachrichten: Die MIH-Funktion kann Nachrichten über MIH-Nutzdaten und das MIH-Protokoll austauschen.

Der Standard beschreibt das MIH-Paketformat, Nachrichtenformate und Prozeduren für den Nachrichtenaustausch um Handover zu erleichtern. Wann ein Handover durchgeführt werden sollte, ist jedoch nicht Teil des Standards.

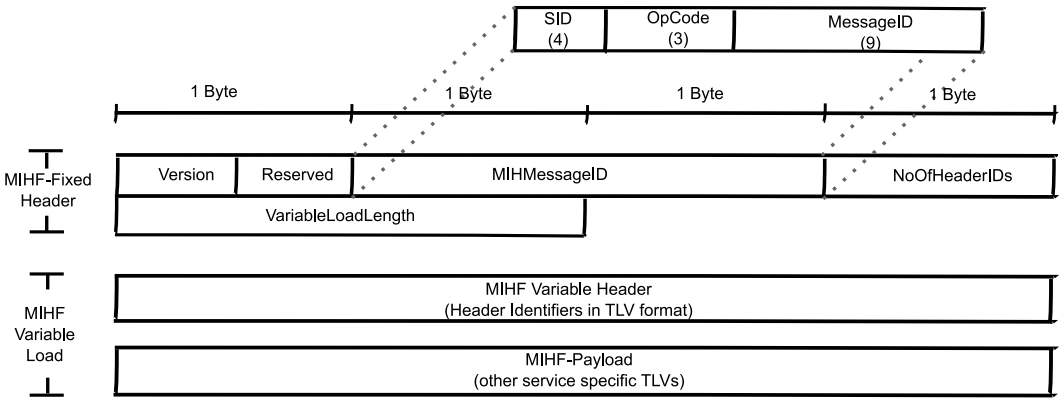

Abbildung 8: Allgemeines MIH-Paketformat

4.2 Allgemeines MIH-Paketformat

Abbildung 8 beschreibt das in [IEEE06a] beschriebene aktuelle MIH-Paketformat. Dabei ist der Paketkopf in zwei Teile unterteilt: den festen Paketkopf und den variablen Paketkopf. Zu dem festen Paketkopf zählen folgende Felder:

- Version: Version des benutzten MIH-Protokolls
- Reserved: Reserviertes Feld (alle Bits auf 0 gesetzt).
- MIHMessageID: Dieses Feld ist eine Kombination aus folgenden Felder:
  - Dienstidentifikator (SIP, 4 Bit): Identifiziert den jeweiligen Dienst (1 für allgemeinen Dienst, 2 für Ereignisdienst, 3 für Befehlsdienst und 4 für Informationsdienst.
  - Operationscode (OpCode, 3 Bit): Typ der Operation (1 für Anfrage, 2 für Antwort und 3 für Anzeige)
  - Nachrichtenidentifikator: (MessageID, 9 Bit): durchzuführende Handlung (z.B. Link Up, Link Going Down, usw.)
- NoOfHeaderIDs: Anzahl der zusätzlichen Identifikatoren im variablen MIHF-Paketkopf.
- VariableLoadLength: Gesamtlänge der variablen Daten, setzt sich aus Länge des variablen Paketkopfes und der Länge der MIHF-Nutzdaten zusammen.

Der variable Paketkopf besteht aus beliebig vielen weiteren Identifikatoren im TLV-Format des MIHF-Paketkopfes und den MIHF-Nutzdaten. Das TLV-Format des MIHF-Paketkopfes ist in Abbildung 9 zu sehen. Ein Beispiel für das Typfeld ist der Transaktionsidentifikator um

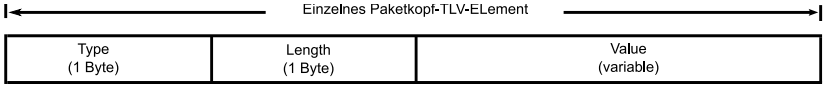

Abbildung 9: MIHF-Paketkopf-TLV-Format

auf Anfragen die jeweilige Antwort zu finden.

Literatur

[IEEE05] IEEE. Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services. IEEE 802.21 Permanent WG Document, Juli 2005. Work in Progress.

[IEEE06a] IEEE. Media Independent Handover Function, Frame Header Contents. , Januar 2006. Work in Progress.

[IEEE06b] IEEE. Media Independent Handover Function, Section 5 Changes. , Januar 2006. Work in Progress.

Abbildungsverzeichnis

Die Entwicklung von 3GPP/UMTS hin zu All-IP

Dominic Jacob

Kurzfassung

Die vorliegende Seminararbeit beschäftigt sich mit der Entwicklung der aktuellen Mobilfunksysteme der dritten Generation (UMTS) hin zu All-IP-Netzwerken. Einleitend wird kurz auf die Standardisierung von Mobilfunksystemen und die daran beteiligten Organisationen eingegangen. Im Idealfall transportiert ein All-IP-Netzwerk (AIPN) möglichst jede Art von Daten paketorientiert auf Basis des IP-Protokolls über eine einheitliche Infrastruktur. Die Architektur der heutigen Mobilfunknetze ist jedoch noch stark vom Nebeneinander der zweiten (GSM) und dritten Mobilfunkgeneration (UMTS) geprägt. Daher wird in Abschnitt 2 zunächst auf die Evolution von 2G-Systemen hin UMTS eingegangen. Dieser skizziert die Anfänge der paketorientierten Übertragung in 2G-Netzen (siehe 2.1) bis hin zur Einführung von UMTS nach dem Release'99 des 3GPP (siehe 2.2). Abschließend werden in Abschnitt 3 einige Beweggründe für die Entwicklung von All-IP dargestellt sowie die bereits erfolgten und noch notwendigen Weiterentwicklungen der Mobilfunkstandards zusammengefasst.

1 Einleitung

Im Dezember 1998 wurde das 3GPP, das 3rd Generation Partnership Project, gegründet. Das Projekt ist ein Zusammenschluss internationaler Standardisierungsorganisationen sowie industrieller Partner. Seine wesentliche Aufgabe ist es global anwendbare technische Spezifikationen für die mobilen Kommunikationssysteme der dritten Generation (3G) zu erstellen.

Diese Kommunikationssysteme basieren auf dem weiterentwickelten GSM-Netzwerk. GSM (Global System for Mobile Communication) hat sich mit weltweit mehr als einer Milliarde Nutzern als der international führende Mobilfunkstandard der zweiten Generation (2G) etabliert. Im Gegensatz zur Entwicklung des GSM-Standards, die mit dem Ziel der Schaffung eines paneuropäischen digitalen Mobilfunknetzwerks Ihren Anfang in Europa nahm, ist 3GPP als globales Projekt angelegt. ETSI, das European Telecommunications Standards Institute, ist als Europäischer Vertreter ein Gründungsmitglied des 3GPP. Weitere beteiligte Institute sind ARIB (Japan), CCSA (China), ATIS (USA), TTA (Korea) und TTC (Japan).

Mit dem IMT-2000 (International Mobile Telecommunications at 2000 MHz), dessen Entwicklung von der ITU (International Telecommunications Union) 1992 angestoßen wurde, konnte ein weltweiter Standard für Mobilfunksysteme der dritten Generation realisiert werden. Dieser ermöglicht neben der Weiterverwendung zentraler Komponenten des GSM-Systems auch eine Evolution bestehender 2G-Netze, die auf dem IS-95-Standard basieren [Spri02]. Dieser ist hauptsächlich in Nord- und Südamerika verbreitet und unterscheidet sich durch das verwendete Multiplexverfahren CDMA (Code Division Multiple Access) wesentlich vom GSM-Standard. UMTS (Universal Mobile Telecommunications System), das Nachfolgesystem von GSM, ist ein Teilstandard der dritten Mobilfunkgeneration und erfüllt somit die Anforderungen des IMT-2000. In Abbildung 1 sind die bei UMTS verwendeten Teilstandards des IMT-2000 hellblau hervorgehoben.

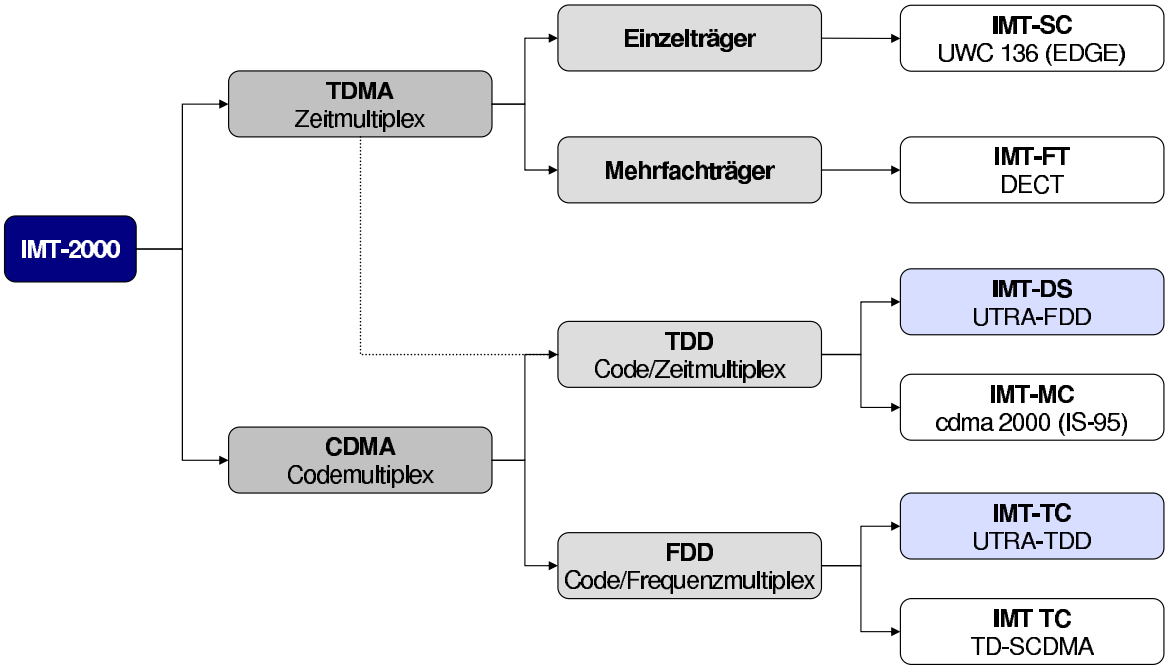

Abbildung 1: Der Standard IMT-2000 in der Übersicht

Das 3GPP hat die erfolgreiche Arbeitsweise bei der Weiterentwicklung des GSM-Standards, der über 20 Jahre fortentwickelt wurde, übernommen und veröffentlicht schrittweise neue Spezifikationen. Jedes so genannte „Release“ stellt dabei eine vollständige Basis für den Aufbau eines 3G-Mobilfunksystems dar. Die heutigen 3G-Netze in Europa basieren in der Regel auf dem Release'99 welches der ITU als Referenz für in den IMT-2000 diente.

Eine wesentliche Rolle bei der Fortentwicklung der 3G-Systeme spielt der zunehmende Bedarf neben Sprachdiensten auch Datendienste effizient über Mobilfunksysteme abzuwickeln. Hiermit sind die Systeme der zweiten Generation überfordert, weil die Netzinfrastruktur auf leitungsorientierte Übertragung ausgelegt ist.

Als Standard für Datendienste bietet sich das TCP/IP-Protokoll an, das dem Internet zu Grunde liegt und einen effizienten Transport von Nutzdaten ermöglicht. Mit GPRS (General Paket Radio Service), einem paketorientierten Übertragungsverfahren, wurde bereits in Hinblick auf die 3G-Systeme eine komplett neue Infrastruktur im Vermittlungsnetz geschaffen. Die erreichbaren Datenraten sind aber auf Grund der Einschränkungen des GSM-Systems immer noch zu niedrig, um mit anderen Zugangstechnologien, wie xDSL (Digital Subscriber Line), zu konkurrieren und neue Dienste zu etablieren.

Das 3GPP hat daher bereits im Release 4 im Jahre 2001 erste Schritte hin zu IP-basierten Netzwerken unternommen und im Release 5 das IP-Multimedia-Subsystem (IMS) sowie HSDPA (High Speed Downlink Packet Access) eingeführt. Release 6 spezifiziert die Zusammenarbeit mit den heute stark verbreiteten WLANs (Wireless Local Area Networks).

Die Entwicklung der 3G-Mobilfunksysteme hin zu vollständig IP-basierten Netzwerken (AIPN - All IP Networks) zeichnet sich also ab. Im Folgenden sollen in Anlehnung an die Machbarkeitsstudie des 3GPP (TR 22.978 V.7.1.0 [Grou05]) die wesentlichen Aspekte dieser Entwicklung skizziert, sowie Akteure, Interessengruppen und wirtschaftliche Gesichtspunkte identifiziert werden.

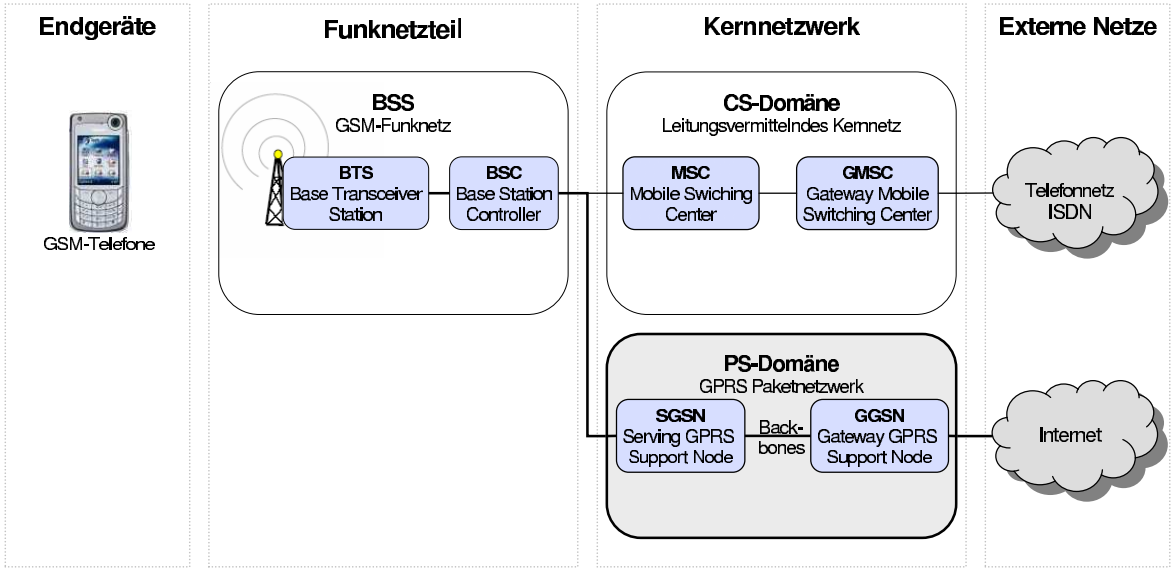

Abbildung 2: Architektur von GPRS/GSM-Systemen

2 2G/3G - Die UMTS Evolution

2.1 2G-Mobilfunksysteme und ihre Weiterentwicklungen

Mit den Mobilfunksystemen der zweiten Generation haben die Betreiber in Westeuropa eine sehr hohe Marktdurchdringung von ca. 90 Prozent erreicht. Der schwerwiegendste Fortschritt für die Konsumenten war die kostengünstige mobile Nutzung von Sprachdiensten. Datendienste allerdings waren bis zum Jahr 2000 lediglich mit leitungsorientierten Übertragungsverfahren möglich. Die CSD-Dienste (Circuit Switched Data) des GSM-Standards erlauben nur Übertragungsraten von bis zu 9,6 kbit/s. Angesichts dieser Einschränkung, versuchten die Netzbetreiber noch vor der Einführung der 3G-Systeme durch Weiterentwicklungen des GSM-Standards zusätzliche Übertragungskapazitäten für den Datenverkehr zu schaffen.

2.1.1 2,5G - HSCSD

HSCSD (High Speed Circuit Switched Data) ist ein beschleunigtes Verfahren mit leitungsorientierter Übertragung. Durch das Zusammenschalten mehrerer Kanäle können Datenraten von bis zu 56 kbit/s erzielt werden. Die Vorteile dieser Technologie sind unter anderem die schnelle Verfügbarkeit, die einfache Realisierung und die konstante Übertragungsrate, die durch die feste Schaltung von Leitungen garantiert wird. HSCSD verursacht allerdings einen hohen Ressourcenverbrauch, weil die reservierten Kanäle für andere Dienste nicht mehr zur Verfügung stehen. Dieser Nachteil und die, im Vergleich zu UMTS, immer noch geringe Übertragungsrate lassen erkennen, dass HSCSD nur eine Übergangslösung auf dem Weg zu den 3G-Systemen sein wird.

2.1.2 2,5G - GPRS

Die Einführung von GPRS Anfang des neuen Jahrtausends war der erste große Schritt in Richtung 3G. Da GPRS paketorientierte Datendienste anbietet, musste von den Netzbetreibern eine eigenständige Infrastruktur im Vermittlungsnetz (Core Network) aufgebaut werden. Abbildung 2 zeigt vereinfacht die wesentlichen Strukturen eines GSM/GPRS-Netzwerks.

Ein IP-basiertes GPRS-Backbone-Netz sorgt für die netzinterne Vermittlung der Datenpakete. Die lokalen Mobilfunkzellen werden mit Hilfe des SGSN (Serving GPRS Support Node) versorgt, das ähnliche Aufgaben wie das MSC (Mobile Switching Center) im leitungsorientierten Vermittlungsnetz wahrnimmt. Durch den GGSN (Gateway GPRS Support Unit) wird eine Anbindung an externe paketorientierte Datennetze wie das Internet ermöglicht. Die Einführung von GPRS ist mit erheblichen Investitionen verbunden. Die logische Trennung zwischen dem leitungsorientierten (CS = Circuit Switched) und dem paketorientierten (PS = Packet Switched) Netzteil wird daher in der ersten Phase der 3G-Systeme beibehalten, da wesentliche Bestandteile der Infrastruktur übernommen werden sollen.

GPRS bietet also erstmals eine vollständig paketorientierte Infrastruktur (PS-Domäne) mit der Daten IP basiert effizient transportiert werden können. Es sind verschiedene Qualitätsstufen für angebotene Dienste (QoS = Quality of Service) verfügbar, die den Bedürfnissen der Anwender angepasst werden können. Zudem können neben Punkt-zu-Punkt Verbindungen auch für die Gruppenkommunikation relevante Punkt-zu-Multipunkt Verbindungen angeboten werden. Ein Nutzer kann demnach an eine definierte Gruppe von Empfängern (Group Call), an Empfänger in einer bestimmten Region (Multicasting) oder an Empfänger die über das Internet angebunden sind (IP-Multicast) Daten versenden [TaBo02].

Die erreichbaren maximalen Datenraten bei GPRS von 45 bis 160 kbit/s können nicht mit aktuellen kabelgebundenen Technologien konkurrieren. Der Flaschenhals liegt dabei weniger im Vermittlungsnetz sondern im Funknetzteil von GSM, das letztendlich nicht auf die paketorientierte Übertragung ausgelegt ist. Im Gegensatz zu den ursprünglichen CSD-Übertragungen, können aber mit GPRS grundlegende Datendienste wie das Abrufen von Mails und Webseiten komfortabel genutzt werden.

2.1.3 2,5G - EDGE

Einen weiteren Evolutionsschritt des GSM-Standards stellt EDGE (Enhanced Data rates for GSM Evolution) dar. Dieses stellt Datenraten von bis zu 384 kbit/s zur Verfügung. Die Steigerung wird letztendlich durch die Übertragung einer größeren Anzahl Bits pro Baut (lineare Modulation im 8PSK-Verfahren) erreicht. Anders als bei GSM repräsentiert hier eine digitale Luftinformationseinheit (Symbol) nicht ein Bit sondern drei Bit [Ahre03]. EDGE macht daher abermals Eingriffe in die Infrastruktur notwendig, wobei diese in erster Linie im Funknetzteil stattfinden. Mit dem E-RAN (Edge Radio Access Network) wird eine weitere vorgelagerte Ausbaustufe hin zum 3G Netzwerk eingeschoben.

Langfristig stellt wohl auch EDGE keine Alternative zum einem weiteren Netzausbau dar. Die gesteigerten Übertragungsraten ziehen aufwändige Fehlerkorrekturverfahren nach sich. Es mussten neun verschiedene Korrekturschemas, deren Einsatz von der verwendeten Bandbreite abhängt, für EDGE entwickelt werden [Tane03]. Die Anfälligkeit für Bitfehler erfordert außerdem eine besonders sorgfältig geplante und gut ausgebaute Funkversorgung , welche gegenüber der GSM-Infrastruktur zusätzliche Investitionskosten verursacht [Ahre03].

2.1.4 2G/2,5G - Zusammenfassung

Zusammenfassend lässt sich feststellen, dass die zweite Mobilfunkgeneration sowie darauf basierende Weiterentwicklungen das Problem des steigenden Datenverkehrs nicht lösen können. Die maximalen Übertragungsraten sind im GSM-Standard auf 384 kbit/s begrenzt (EDGE) und wenig flexibel anpassbar. In der Praxis werden meist nur weit geringere Raten erzielt. EDGE erfordert wegen seiner Fehleranfälligkeit eine hohe Qualität der Funkversorgung, die nur in seltenen Fällen vorhanden sein wird [Kroe04]. GPRS bietet maximal 13,2 kbit/s auf

einem Kanal, wobei bis zu acht Kanäle gebündelt werden und sich maximal acht Anwender einen Kanal per Zeitmultiplex teilen. Die Anzahl der verfügbaren Kanäle, und damit auch die Übertragungsrate in einer Funkzelle, wird von der Anzahl der eingebuchten Teilnehmer und der Datenaufkommen beeinflusst. In Gebieten mit starker Netzauslastung führt das zwangsläufig zu geringeren Übertragungsraten. Überdurchschnittlich lange Paketumlaufzeiten von bis zu 1000 ms (via xDSL ca. 65 ms) verlangsamen Aufrufe von Internetseiten stark und relativieren die gesteigerten Bandbreiten. Allen genannten Techniken ist außerdem gemein, dass durch Ihren Einsatz die Kapazität des GSM-Netzwerks verringert wird [Ahre03].

Aus Sicht der Endanwender spricht vor allem das im Vergleich zu anderen Zugangssystemen wie WLAN (Wireless Local Area Network), xDSL oder ISDN (Integrated Services Digital Network) extrem hohe Preisniveau gegen eine intensive Nutzung der Mobilfunknetze für den Datenverkehr. Während eine Minute Internetzugang per WLAN-Hotspot mit ca. 1,5 Cent pro Minute taxiert wird, sind für GSM-Verbindungen Beträge in der Größenordnung von etwa 30 Cent üblich [Riem03]. Auch per Übertragungsvolumen abgerechnete Tarifmodelle sind für die Anwender bisher nicht attraktiv genug.

2.2 3G/UMTS - Die erste Ausbaustufe

Die Einführung der dritten Mobilfunkgeneration, die in Europa seit Ende 2003 angelaufen ist, verspricht viele der oben genannten Probleme zu lösen. So standen zum Beispiel der GSM-Funknetzteil mit den verwendeten Zeitmultiplexverfahren und die stark auf leitungsvermittelnde Dienste ausgelegte Infrastruktur im Kernnetzwerk einer effizienten paketorientierten Vermittlung stets im Wege. Da die Umstellung in Europa im Gegensatz zu einigen asiatischen Ländern schrittweise auf Basis der vorhandenen GSM-Infrastruktur erfolgt, werden auch neue Dienste und gesteigerte Übertragungsraten erst nach und nach zur Verfügung stehen.

2.2.1 Überblick

Die erste Ausbaustufe der 3G Mobilfunksysteme in Westeuropa basiert auf dem Release'99 des 3GPP. Die einschneidenden Neuerungen gegenüber dem GSM/GPRS-Systemen betreffen das Funknetzteil. Dort wird eine vollständig neue Infrastruktur namens UTRAN (UMTS Terrestrial Radio Access Network) eingeführt. Die Komponenten der 2G-Architektur im Vermittlungsnetzwerk werden im Wesentlichen weiterverwendet und auf die neuen Anforderungen hin ausgebaut. Die Koexistenz der GSM- und UMTS-Systeme wird in Abbildung 3 dargestellt.

2.2.2 Das Funknetzteil UTRAN

Der Funknetzteil der dritten Generation setzt im Unterschied zu GSM auf ein Codemultiplexverfahren. Das verwendete WCDMA-Verfahren (Wideband Code Division Multiple Access) ermöglicht erheblich höhere Übertragungsraten als bei GSM und ist durch die Spreizung des Signals auf 5 MHz (GSM 200 kHz) zudem weniger störungsanfällig.

Die Netzbetreiber konnten zum Beispiel in Deutschland Frequenzpakete ersteigern, die aus bis zu zwei gepaarten 5MHz Frequenzen (Up- und Downlink) bestehen und außerdem zusätzlich eine einzelne ungepaarte Frequenz beinhalten. Zwei gepaarte Frequenzpakete werden dabei als Mindeststandard für einen effizienten Netzbetrieb erachtet. Auf den gepaarten Frequenzen wird synchron mit dem UTRA-FDD-Modus (Frequenz Division Multiplex) übertragen, während die einzelne Frequenz im UTRA-TDD-Modus (Time Divison Multiplex) für asynchrone Übertragungen genutzt wird. Diese eignet sich somit vornehmlich für Datenübertragungen.

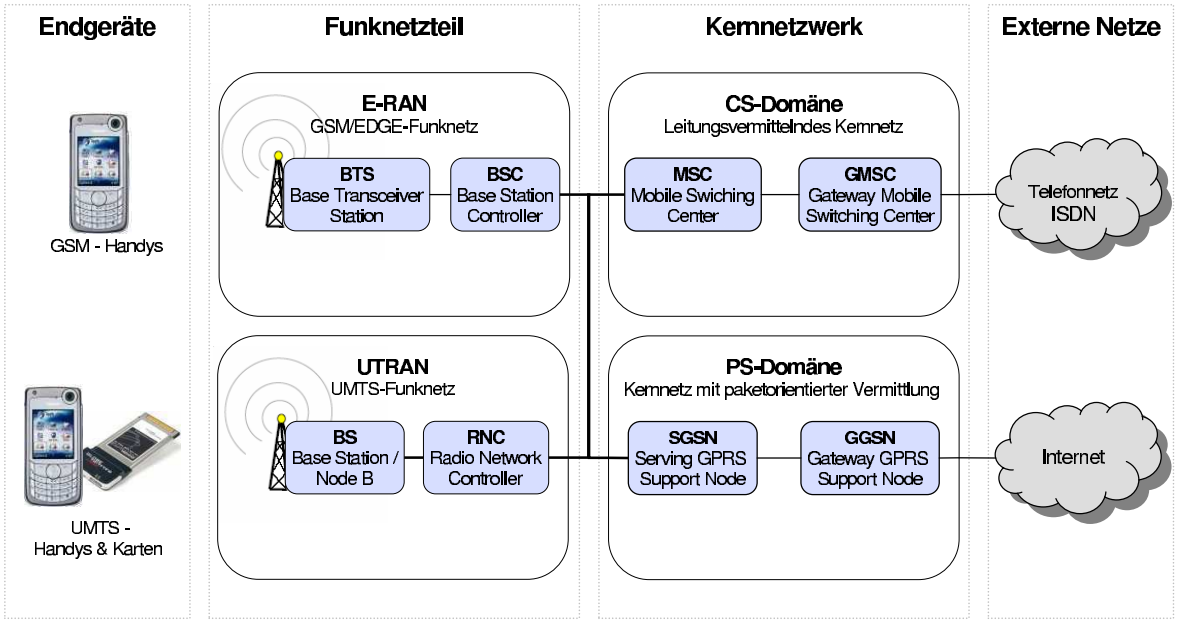

Abbildung 3: Architektur von UTMS-Systemen nach Release'99

Die Topologie des UTRAN erlaubt Datenraten von bis zu 2 Mbit/s, die anfänglich nur im asynchronen UTRA-TDD-Modus erreicht werden können. Die verfügbare Übertragungsrate hängt von der Geschwindigkeit des Anwenders und von seiner Position im Versorgungsgebiet ab. Das Netzwerk ist dabei in verschieden große Funkzellen unterteilt, die sich gegenseitig überlagern. Makrozellen sind großräumig dimensioniert (ca. 2 km) und bieten Datenraten von bis zu 144 kbit/s. Mikrozellen haben einen Durchmesser von ca. 1 km, bieten bis zu 384 kbit/s und sind vornehmlich in Ballungsräumen eingerichtet. Pikozellen haben eine sehr geringe Ausdehnung von rund 60 m und sind zukünftig in erster Linie als Hotspots in Flughäfen, Bahnhöfen und ähnlichen öffentlichen Gebäuden zu finden. Soweit in einem Bereich mehrere Funkzellen (z.B. Makro- und Mikrozelle) verfügbar sind, können sich Endgeräte je nach Anforderung (z.B. Dienstgüte) in eine der Zellen einbuchen und bei veränderten Bedingungen gegebenenfalls wechseln. Der UTRA-TDD-Modus, und somit auch die maximale Datenrate von 2 Mbit/s, ist vorläufig lediglich für den Einsatz in Pikozellen vorgesehen. Flächendeckend stehen mit der Einführung von UMTS also Datenraten von 144 bzw. 384 kbit/s zur Verfügung [Cast04].

UTRAN hat ähnliche Komponenten wie das Funknetzteil von GSM. Die Node B (Bodenstation) versorgt meist drei Funkzellen und hat ihr Äquivalent im GSM-Netz in der BTS (Base Transceiver Station). Der RNC (Radio Network Controller) findet sein Gegenstück im BSC (Base Station Controller) des GSM-Netzes (vgl. Abbildung 3). Die RNC sind untereinander verbunden und können so, anders als bei GSM wo diese Aufgabe im Kernnetzwerk abgewickelt wird, selbständig einen Teilnehmer an eine andere Funkzelle übergeben (Handover), auch wenn diese Zelle von einem anderen RNC bedient wird. In Überlappungsgebieten zweier Funkzellen werden gleichzeitig zwei Funkkanäle aufgebaut, auf denen jeweils das gleiche Signal (Splitting) übertragen wird. Dieses wird dann im RNC wieder zu einem möglichst störungsfreien Signal zusammengesetzt (Combining). Dadurch soll kompensiert werden, dass sich die Sendeleistung der Endgeräte in Randgebieten stark erhöht, was beim WCDMA-Verfahren zu einer gesteigerten Fehlerrate führt. Für den Handover-Vorgang hat das den Vorteil, dass beim Wechsel der Zelle bereits ein Kanal besteht und somit die Verbindung nicht unterbrochen wird (Softhandover) [Riem03].

2.2.3 Das Vermittlungsnetzwerk (Core Network)

Das Vermittlungsnetzwerk ist weiterhin in zwei logische Bereiche aufgeteilt. Die CS-Domäne (Circuit Switched Domain) übernimmt die Verwaltung von leitungsorientierten Diensten, basiert auf der GSM-Architektur und verwendet einen Großteil ihrer Komponenten. Die PS-Domäne (Packet Switched Domain) baut auf dem GPRS Core Network auf, ist aber anders als bisher, direkt an Radio Network Controller (RNC) des UTRAN angebunden. Durch diese direkte Anbindung können kürzere Paketumlaufzeiten und schnellere Übertragungsraten erzielt werden. Die Anbindung der beiden Funknetzteile und der CS- bzw. PS-Domänen untereinander erfolgt über ein leistungsfähiges ATM- oder IP- Backbone-Netz.

2.2.4 Zusammenfassung

Die erste Ausbaustufe der UMTS-Mobilfunknetze auf Basis von IMT-2000 bzw. dem 3GPP Release'99 stellt einen Kompromiss zwischen dem nachvollziehbaren Bedürfnis vorhandene Infrastrukturen weiterzubetreiben und dem Ausbau hin zu einem vollständig paketorientierten 3G-System dar. Die Kapazität des UMTS-Netzes beträgt beim vollständigen Aufbau in etwa das Fünffache des GSM-Netzes [Kroe04]. Geeignete Endgeräte können durch zuverlässige Handover-Verfahren je nach Verfügbarkeit zwischen GSM- und UMTS-Netz umschalten, ohne dass die Verbindung abbricht. Auch stehen verschiedene QoS-Klassen zur Verfügung, die in Abhängigkeit vom verwendeten Dienst oder dem Nutzerstatus (z.B. Gold-, Silber, Bronzenutzer) dynamisch angepasst werden können. Die angestrebte Datenrate von 2 Mbit/s in den Pikozellen, die durch den UTRA-TDD-Modus auf dem ungepaarten 5 MHz Band erreicht werden soll, bleibt bis zum Aufbau einer großräumigen Abdeckung mit Makro- und Mikrozellen wohl eher die Ausnahme.

Für den effizienten Transport von Nutzdaten bedarf es dennoch eines weiteren Ausbaus dieser Architektur. 3GPP hat daher weitere Spezifikationen sowie das All-IP-Konzept entwickelt, die im Weiteren näher dargestellt werden.

3 Das All-IP Konzept

In der All-IP Network (AIPN) feasibilty study (TR 22.978 V.7.1.0) [Grou05] skizziert das 3GPP die weitere Entwicklung der 3GPP-Systeme hin zum All-IP Netzwerk. Im Folgenden werden die wesentlichen Aussagen der Studie kurz dargestellt. Des Weiteren soll ein kurzer Überblick zu bereits standardisierten Weiterentwicklungen der 3GPP-Spezifikationen die erfolgten Schritte hin zu All-IP nachvollziehen.

3.1 Gründe für All-IP

Bei der Entwicklung des AIPN ist es ein vorrangiges Ziel, auf Basis des IP-Protokolls, eine universelle und nahtlose Zugriffsmöglichkeit von verschiedenen Netzwerken aus zu schaffen. Weiterhin wird eine verbesserte Anwenderfreundlichkeit angestrebt, die sich zum Beispiel in höheren Übertragungsraten, schnelleren Einbuchungen und konstant guter Qualität ausprägt. Die Netzbetreiber erwarten sich außerdem eine Reduktion der Kosten sowohl für den Betrieb als auch für die Ausrüstung und den Ausbau des Netzwerks. Die Flexibilität bei der weiteren Evolution der 3G-Netze stellt ebenfalls einen wichtiger Faktor dar. Ein AIPN soll es den Betreibern ermöglichen, selbständig neue Dienste zu entwickeln und weiterhin gewachsene Infrastrukturen wie beispielsweise die CS-basierten Domänen zu nutzen.

3.1.1 Motivation aus Sicht der Anwender

Diversifikation von mobilen Diensten

Der allgemeine Konsumtrend geht hin zu einer weiteren Diversifikation von mobilen Diensten, d.h. es soll in Zukunft möglich sein besser auf die individuellen Bedürfnisse von einzelnen Anwendern oder Anwendergruppen einzugehen. Neben dem traditionellen Anwendungsmuster Server-zu-Nutzer, können vielfältige Szenarien für die Kommunikation von Nutzer-zu-Nutzer (Chat, Ad-hoc Netze u.a.) entwickelt werden. Mit Hilfe variabler Dienstgüten (QoS) können zudem unterschiedliche Qualitätsansprüche der Anwender berücksichtigt werden. Schließlich soll es möglich sein, je nach verwendetem Endgerät, angepasste Dienste anzubieten und einen weitestgehend nahtlosen Zugriff mit verschiedenen Endgeräten und Zugangsnetzen sicherzustellen.

Interaktion mit neuen Endgeräten

In den meisten Industrieländern hat der Mobilfunk mit Handys eine sehr hohe Marktdurchdringung erreicht. Zusätzlich geht die Entwicklung dahin, dass Nutzer mit mehreren Endgeräten (bis hin zu Kühlschränken etc.) über das Netzwerk kommunizieren wollen. Dies ist mit der aktuellen Limitierung durch den Telefonnummernraum (MSISDN) nicht realisierbar. Ein AIPN würde hier für eine Vielzahl von Endgeräten eine Adressierung ermöglichen (IPv6).

3.1.2 Motivation aus Sicht der Netzbetreiber

Datenvolumen übersteigt Volumen von Sprachdiensten

Zukünftige Mobilfunknetzwerke müssen in der Lage sein wesentlich mehr paketorientierten Verkehr zu verarbeiten als das bisher der Fall war. Das Volumen wird in absehbarer Zeit den durch Sprachdienste (CS) verursachten Verkehr übersteigen, was einen weiteren Ausbau der PS-Domäne sinnvoll erscheinen lässt. Das hätte außerdem den Vorteil, dass IP-Daten für verschiedenste Anwendungszwecke (Nutzer-zu-Nutzer, Multicasting etc.) auf diese Weise effektiver und somit kostengünstiger transportiert werden.

Unterstützung von verschiedenen Zugangssystemen

Mit dem Release 6 hat das 3GPP bereits die Möglichkeit geschaffen, über WLAN-Netzwerke auf das 3GPP-System zuzugreifen. Es ist zu erwarten, dass die Netzbetreiber und Dienstanbieter zukünftig noch weitere Zugangsmöglichkeiten schaffen wollen. Hierfür wird ein einheitliches Zugangssystem benötigt, dass kosteneffektiv Zugriff über verschiedene Luftschnittstellen erlaubt. Ein AIPN könnte den Zugriff von vielen Zugangsnetzen aus unterstützen und würde dabei nur minimale Anforderungen an die externen Systeme stellen. Das IP-Protokoll als Basis zu nutzen, erlaubt den Betreibern außerdem allgemein verbreitete IP-Technologien einzusetzen und eine einheitliche IP-Schnittstelle zu entwickeln.

Verbindung der IT- und Telekomwelt

Mit dem überproportionalen Wachstum des breitbandigen Internets ist die Zahl der Nutzer des IP-Protokolls stark angestiegen. Auch die IP-Telefonie erfreut sich wachsendem Zuspruchs. Für die Systeme des 3GPP ist es von zunehmender Bedeutung diesen Trend zu folgen und vergleichbare Angebote zu entwickeln. Des Weiteren ist es sinnvoll die Zusammenarbeit mit VoIP-Anbietern zu ermöglichen, da mit der stark steigenden Anzahl der VoIP-Nutzer die Kommunikation zwischen den Netzen zunehmen wird.

Während vollständig standardisierte Systeme der zweiten Generation (GSM) in der IT-Welt kaum eine Rolle spielten, bieten die 3G-Systeme durch weniger stark regulierte Dienste weiterreichenden Spielraum zur Entwicklung neuer Anwendungen auf Basis einer einheitlichen

Plattform. Es hat sich gezeigt, dass durch eine weniger starke Standardisierung im Detail, Entwicklungen von der Industrie schneller vorangetrieben werden. Die Vielfalt und das starke Wachstum von Internetdiensten sind ein überzeugendes Beispiel für den Erfolg dieses Ansatzes. Die IP-Technologie bietet für die Industrie also den Vorteil, dass mit Ihr neue Dienste auf Basis von defacto Standards wesentlich schneller realisiert werden können.

Kostenreduktion bei Ausrüstung (CAPEX) und Betrieb (OPEX)

Der zunehmende Druck mit anderen Anbietern, insbesondere IP-Service-Anbietern, zu konkurrieren, erfordert einen effizienten Ausbau der 3GGP-Systeme, der kostengünstig wesentlich höheren IP-Verkehr erlauben sollte. Auch ohne Berücksichtigung dieses Wettbewerbs, ist mit einer erheblich steigenden Menge an Datenverkehr zu rechnen. Ein Ausbau der Infrastruktur ist daher unumgänglich. Durch den einheitlichen Einsatz von IP-Technologien können Kosten für diesen Ausbau eingespart werden, weil der Markt im Gegensatz zu spezialisierten Lösungen sehr groß ist. Die Technologien können außerdem mit vergleichbar geringem Aufwand an die Bedürfnisse der Netzbetreiber angepasst werden.

3.1.3 Motivation aus technischer Sicht

Evolution für die nächste Mobilfunkgeneration

Ähnlich zum Fortschritt von 2G auf 3G ist auch für zukünftige Systeme eine signifikant höhere Übertragungsrate zu erwarten. Es erscheint also sinnvoll Systeme so auszulegen, dass verschiedene Aspekte, wie Dienstgüte etc., auf die Bedürfnisse der Nutzer beziehungsweise der verwendeten Dienste angepasst werden können. Die Architektur eines AIPN kann bereits Anforderungen für die nächste Generation von Mobilfunksystemen berücksichtigen und so die weitere Evolution fördern.

Aufkommen neuer Technologien

Die stark zunehmende Verbreitung von WLAN-Netzwerken und vergleichbaren Technologien, die relativ hohe Bandbreiten zu günstigen Preisen anbieten, legt es nahe, eine Zusammenarbeit mit diesen Systemen weiter zu fördern (vgl. Release 6). Auch Aktivitäten um Ad-hoc Netzwerke finden derzeit noch außerhalb der 3G-Netze statt. Für die Netzbetreiber ist es eine vielversprechende Perspektive, bei der Möglichkeit zur Interaktion mit solch spontanen Netzwerken, neue Dienste zu entwickeln. Wichtige Aspekte, die dabei zu berücksichtigen sind, wären zum Beispiel die Identifikation und Einbuchung der Nutzer sowie die korrekte Adressierung und Routing der verwendeten Endgeräte.

Neben WLAN-Netzwerken (Infrastruktur, Ad-hoc) werden sich in den nächsten Jahren weitere neue Zugangstechnologien etablieren, wie zum Beispiel RFID, Personal Networks oder Multi-hop Access Networks, die auf paketorientierter Vermittlung basieren. Auch Webservices gewinnen in der Industrie zunehmend an Bedeutung. Dieses Ertrags- und Synergiepotenzial könnte mit einem AIPN, das für neue Technologien vorbereitet ist, erschlossen werden.

Zugriff über verschiedene Luftschnittstellen

Ein AIPN kann für die Nutzung verschiedener Zugangssysteme in einem Areal eine einheitliche Plattform zur Verfügung stellen. Der Wechsel eines Nutzers (Handover) von einem in das andere Netz soll dabei einfach (für den Nutzer) und reibungslos von statten gehen. Möglich sind auch Szenarien, in denen ein Nutzer über mehrere Netzwerke gleichzeitig zugreift. Ein AIPN würde dabei auch die dynamische Regelung von Bandbreiten, Dienstgüte und Tarifen ermöglichen. Des Weiteren können Zugriffsregeln und Sicherheitsstandards durch die Netzbetreiber festgelegt werden.

Fortgeschrittene Datenflussoptimierung

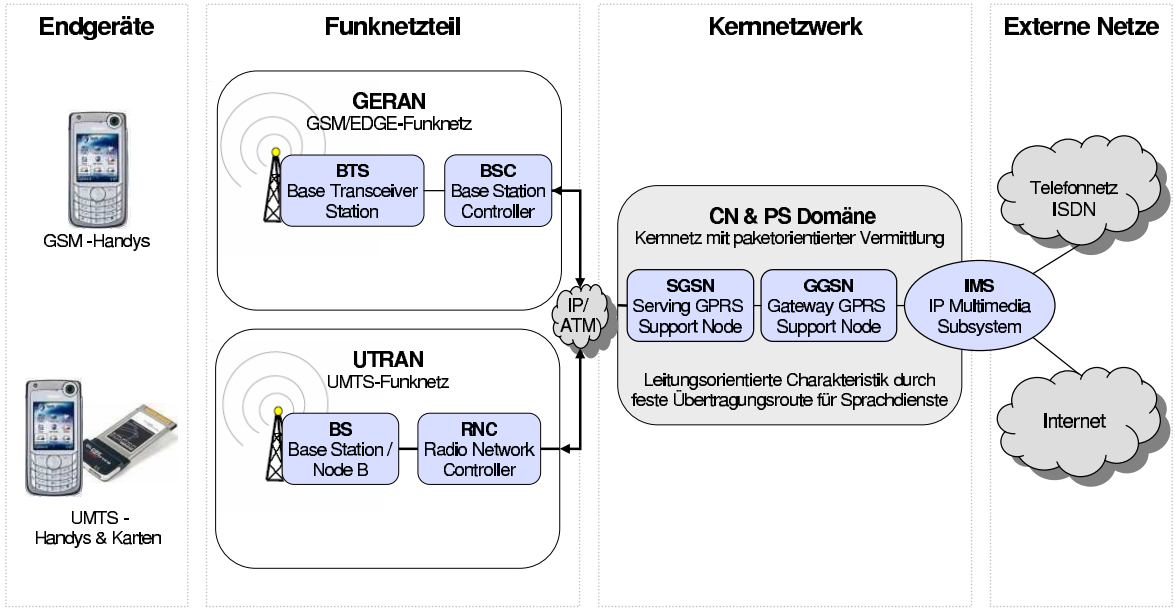

Abbildung 4: Architektur von UTMS-Systemen nach Release 4/5

Auch im AIPN sind durch den gesteigerten Datenverkehr gelegentliche Flaschenhälse im Backbone-Netz der Betreiber zu erwarten. Bereits heute existieren IP-Technologien, die für diese Problemstellung Lösungsansätze bieten. So können besipeilsweise fortgeschrittene Routing Algorithmen, MLPS und dynamische Lastkontrolle helfen einen effizienten Netzbetrieb zu gewährleisten.

3.2 Die ALL-IP Evolution bis zum Jahr 2005

Wichtige Schritte auf dem Weg zum AIPN wurden, zumindest auf Seiten der Standardisierungsorganisationen, bereits in den letzten Jahren gemacht. Das 3GPP hat seine Spezifikationen seit dem Release'99 (vgl. Abb. 3) mehrfach weiterentwickelt. Die Umsetzung durch die Netzbetreiber hängt dieser Entwicklung allerdings noch hinterher.

3.2.1 2001 - Release 4

Im Release 4 wird die Struktur der CS-Domäne durch die Einführung des MSC-Servers sowie von Mediagateways (MGW) grunderneuert. Das leitungsorientierte CS-Kernnetzwerk wird durch diese Komponenten ersetzt. Der MSC-Server ist nicht mehr, wie vormals die MSCs (Mobile Switching Center), für die Schaltung von Leitungen zuständig, sondern dient als Call Server. Er übernimmt die Steuerung von Sprachdiensten, die über die Mediagateways in andere Netze beziehungsweise Netzteile weitergegeben werden. Die CS-Infrastruktur ist also auf eine paketorientierte Übertragung umgestellt. Für die Sprachdienste wird eine Übertragungsroute im Netzwerk definiert, die lediglich eine leitungsorientierte Charakteristik hat [Riem03].

3.2.2 2002 - Release 5

Mit dem Release 5 wird das IP Multimedia Subsystem (IMS) eingeführt und die CS- endgültig mit der PS-Domäne zusammengelegt (vgl. Abb. 4). Auch der Datenverkehr zwischen den

Kernnetz und den Funknetzteilen, sowie zwischen Funkstationen und Funkkontrolleinheiten (RNC, BSC) wird vollständig auf ein IP- oder ATM-basiertes Backbone-Netz umgestellt.

Das IMS stellt eine neue universelle Dienstplattform dar, die vielfältige Multimedia-Anwendungen auf Basis des All-IP-Konzepts ermöglichen soll. IMS basiert auf dem Session Initiation Protocol (SIP) und weiteren Protokollen, die von der IETF (Internet Engineering Taskforce) standardisiert wurden. SIP ist auch das in der Regel von VoIP-Anbietern (Voice over IP) genutzte Signalisierungsprotokoll. Das IMS fungiert zudem als zwischengeschaltete Instanz zwischen dem Kernnetzwerk und externen Netzen wie dem Internet und kabelgebundenen Telefonnetzen. Durch den mehrschichtigen Aufbau der IMS-Architektur (vgl. Abb. 5), wird eine einfachere Einführung und Integration von neuen Diensten möglich. Das IMS bietet Anwendungen ein Framework an, das es erlaubt so genannte Service Enablers zu nutzen. Beispiele hierfür wären u.a. digitale Rechteverwaltung (DRM = Digtal Rights Management), Nachrichtendienste und Push to Talk over Cellular (PoC) [ScAS05].

Ein Kernelement der IMS-Infrastruktur ist die Call Session Control Function (CSCF), die als SIP Server Aufgaben im Call- und Session-Management, einschließlich Authentifikation, erfüllt. Home Subscriber Server (HSS) sind zuständig für Benutzer-Identifikation und Zugangs-Autorisierung sowie für Verwaltung von Dienst- und Kundenprofilen. Die Multimedia Resource Function (MRF) ermöglicht zusätzliche Dienste wie Konferenzschaltung, Interactive Voice Response Services (IVR) und Medientranskodierung. Media Gateways (MGW) dienen der Umwandlung von Media-Codecs und Protokollen zum Beispiel beim Übergang von leitungsvermittelten und paketorientieren Netzen. Schließlich bieten die IMS Application Server die SIP-basierten Dienste an [ScAS05].

Zusammenfassend lässt sich feststellen, dass mit der Einführung des IMS ein großer Schritt hin zum All-IP Netzwerk gemacht wird. Vielfältige Dienste lassen sich einfach und kostengünstig von anderen IP basierten Infrastrukturen, wie dem Internet, adaptieren. Bei neuen Anwendungen werden die Anbieter von Synergieeffekten, die durch die einheitliche Entwicklungsplattform entstehen, profitieren.

Eine weitere Neuerung mit dem Release 5 ist das Verfahren HSDPA (High Speed Downlink Packet Access), das Bruttodatenraten von bis zu 14,4 Mbit/s verspricht. Die Steigerung der Kapazität der Funkstationen (Node B) wird durch den Einsatz eines neuen Modulationsverfahrens (16QAM) sowie eine effizientere Lastverteilung erreicht. Wie der Name bereits aussagt, wird diese verbesserte Übertragungsrate nur für Downloads aus dem UMTS-Netz zur Verfügung stehen. Eine Variante, die auch den Upload beschleunigt (HSUPA - High Speed Uplink Packet Access), wird bereits vom 3GPP entwickelt.

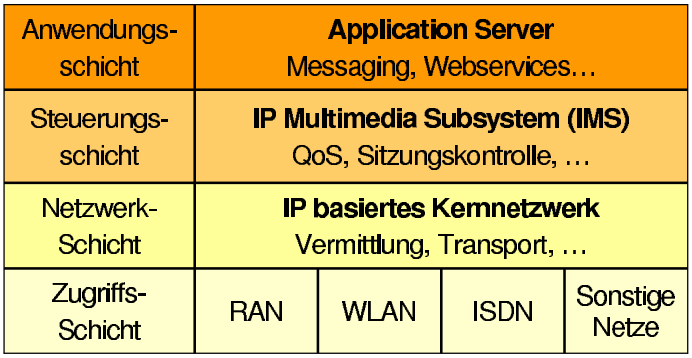

Abbildung 5: Schichtenweiser Aufbau des IP Multimedia Subsystems

3.2.3 2004 - Release 6

Das Release 6 bringt für die Infrastruktur des 3GPP-Systems keine schwerwiegenden Neuerungen. Es definiert die Anforderungen für eine Zusammenarbeit der 3G-Netze mit dem weit verbreiteten Netzwerk-Standards (u.a. WLAN) und ermöglicht den Zugang per WLAN als ein alternatives Funknetz. Des Weiteren werden einige Verbesserungen am IMS sowie Optimierungen am Funknetzteil implementiert.

3.3 Die weitere Entwicklung von All-IP

Die weitere Entwicklung der 3GPP-Systeme, mit dem Ziel eines vollständigen All-IP Netzwerks, soll auf den bisherigen Ausbaustufen aufsetzen. In den vorhergehenden Releases sind bereits viele Strukturen definiert worden, die für ein kommendes AIPN nutzbar sind (siehe Abschnitt 3.2). Für die Zukunft sind dennoch an einigen Stellen Verbesserungen notwendig oder zumindest sinnvoll.

3.3.1 Kernelemente des AIPN

Das 3GPP identifiziert in seiner Studie einige Schlüsselelemente, die bei der Entwicklung eines AIPN vorrangig berücksichtigt werden sollen [Grou05]. Abbildung 6 zeigt in einer groben Übersicht den idealisierten Aufbau eines AIPN.

IP-Netzwerk, Performance, Routing und Adressierung

Grundsätzlich wird eine weitere Steigerung der Netzwerk-Performance angestrebt. Die bisherige Struktur war auf den Hauptanwendungsfall Nutzer-zu-Server hin konstruiert worden. Ein zukünftiges Netz sollte alternative Anwendungsfälle (Nutzer-zu-Nutzer, Multicasting, etc.) berücksichtigen, weithin optimiertes Routing einsetzen und besonders im Funknetzsteil vorhandene Ressourcen effizient nutzen. In Folge der Etablierung von IP-Technologien in der Industrie, ist es naheliegend Routing und Adressierung zukünftig IP-basiert abzuwickeln, um eine Anpassung an eine möglichst große Anzahl von Nutzern und Endgeräten zu ermöglichen. Die komplette Vermittlungsinfrastruktur soll also im Endergebnis auf IP-basierte Übertragung umgestellt werden.

Unterstützung von Zugangssystemen

Eine weitere Öffnung des Kernnetzes für Zugriffe von externen Netzwerken ist unumgänglich, um Übergänge (Handover) zu ermöglichen und den Betreibern neue Ertragspotentiale zu erschließen. Dabei ist es wichtig ein übergreifendes Abrechnungssystem zu entwickeln und Dienstmerkmale über die verschiedenen Systeme hinweg einheitlich anzubieten. Anknüpfend an diese Problematik sollten Möglichkeiten geschaffen werden von mehreren Netzwerken aus gleichzeitig zuzugreifen, Zugangssysteme zu wählen und die Dienste an die verwendete Zugangsoption anzupassen, wobei die Steuerung und Reglementierung dabei den Netzbetreibern überlassen wird.

Mobilität und Sitzungskontrolle

Um mobile Sitzungen und Sitzungsübergaben an Terminals oder andere Endgeräte zu unterstützen, ist es notwendig die vorhandenen Steuerungsmechanismen des IP Multimedia Subsystems weiter zu entwickeln. So soll es zum Beispiel möglich sein, gesteuert durch den Netzbetreiber, eine aktive Sitzung an ein anderes Endgerät weiterzugeben und diese wenn nötig auf die dortigen Gegebenheiten anzupassen. Auch für Multicast-Sitzungen soll eine Steuerungsoption entwickelt werden. Ein weiteres Ziel in diesem Zusammenhang ist es Nutzerpräferenzen, Prioritäten, Netzwerkeinstellungen und sonstige für die Betreiber relevante

Kriterien in Zukunft flexibler steuern zu können. Schließlich ist eine entsprechende Skalierbarkeit der Anwendungen zu berücksichtigen. Die Nutzer sollen also einerseits zuverlässigen Zugriff auf die Dienste erhalten, während auf der anderen Seite eine effiziente Nutzung von Funk- und Energieressourcen sichergestellt wird.

Dienstgüte (QoS)

UMTS bietet gegenüber GSM bereits stark verbesserte Möglichkeiten die Dienstgüte flexibel anzupassen. Bei der Entwicklung des AIPN ist es dennoch sinnvoll, wenn die Netzbetreiber zukünftig in die Lage versetzt werden die Dienstgüte dynamisch, in Abhängigkeit von den zur Verfügung stehenden Ressourcen, anzupassen. Hierfür ist es notwendig die aktuelle Auslastung sowie unbelegte und reservierte Ressourcen in Echtzeit zu überwachen und Datenverkehr dynamisch durch das Netzwerk zu routen. Ein AIPN sollte im Idealfall für verschiedene Arten von IP-Verkehr variable QoS-Klassen anbieten und Mechanismen für die Netzbetreiber zur Verfügung stellen, um diese je nach Netzlast einzusetzen. Um die genannten Ziele zu erreichen, können zum Beispiel dynamische Lastverteilungs- und Routing-Algorithmen implementiert werden.

Neue Dienste

Mit dem Ausbau zum AIPN wird die Entwicklung neuer Anwendungen möglich. Die Einführung des IMS (vgl. Abschnitt 3.2) hat hier bereits die Grundlagen für die Entwicklung einer fexiblen Dienstplattform geschaffen. In der Zukunft kann diese Infrastruktur noch weiter verbessert werden, indem das Konzept der Application Server fortentwickelt wird und für die Gruppenkommunikation neue Service Enabler, wie zum Beispiel Video Group Call, angeboten werden. Auch die Kombination verschiedener Dienste wie SMS, MMS und Instant Messaging stellt eine attraktive Option dar.

Sicherheit und Datenschutz

Grundsätzlich muss ein AIPN mindestens den derzeitigen Sicherheitsanforderungen der 3GGP-Systeme genügen. Wünschenswert wäre, gerade in Anbetracht der Öffnung für externe Zugangssysteme, eine zuverlässiger Schutz der netzinternen Komponenten vor Zugriffen außerhalb des AIPN. Der grundlegende Schutz der Privatspäre der Anwender muss ebenfalls gewährleistet werden. So sollten zum Beispiel Datenübertragungen durch geeignete Verschlüsselungsverfahren (u.a. IPSec) geschützt werden.

Einflüsse auf das 3GPP-System

Für die endgültige Einführung des AIPN sind auch in Zukunft noch Eingriffe in die Infrastruktur der 3GPP-Systeme notwendig. Die Evolution soll dabei ausgehend von Release 6 fortgeführt werden, dass heißt soweit möglich auf den vorhandenen Strukturen aufbauen und Abwärtskompatibilität sicherstellen. Der Fokus bei der Entwicklung soll auf der Verbesserten Unterstützung von Mobilität und alternativen Zugangsnetzen liegen. Wobei es nicht das vorrangige Ziel ist, über alle Fremdnetze das komplette Angebot der eigenen Netze GERAN und UTRAN verfügbar zu machen. Zudem muss eine Koexistenz der AIPN- und der vorhandenen Funktionen zum Mobilitätsmanagment in der PS-Domäne möglich sein.

3.4 Abschließende Bewertung

3.4.1 Perspektiven

Das 3GPP hat in der Machbarkeitsstudie (TR 22.978) [Grou05] vielfältige Anforderungen identifiziert, die in den weiteren Entwicklungsprozess hin zum All-IP Netzwerk berücksichtigt werden müssen. Mit dem Release 7 soll eine genaue Spezifikation der Dienstanforderungen

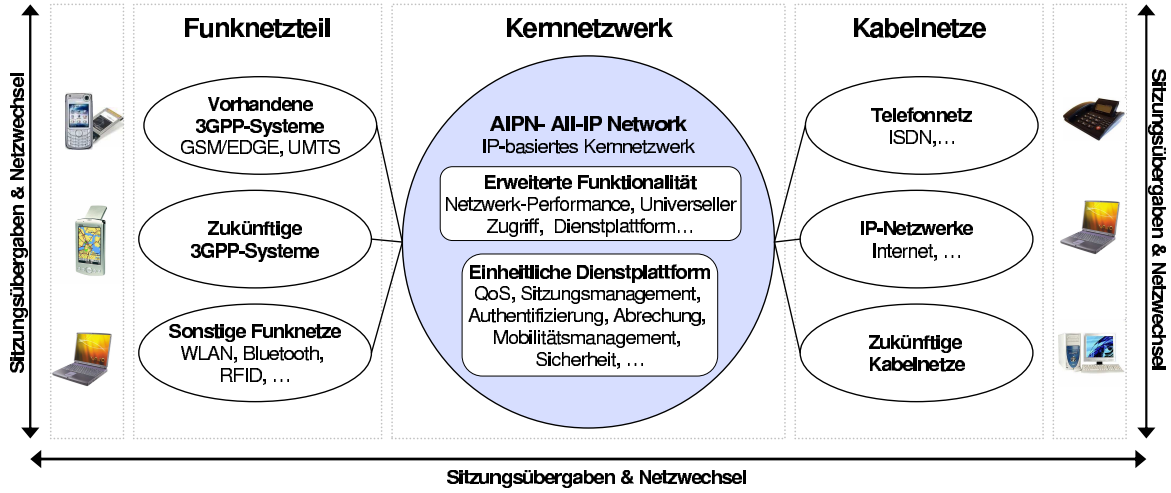

Abbildung 6: Vision des All-IP Netzwerks

für das AIPN veröffentlicht werden. Die Einführung neuer und die Verbesserung vorhandener Funktionalitäten werden von 3GGP als notwendig angesehen, um die Schlüsselelemente eines AIPN zu verwirklichen. Die Entwicklung dieser Spezifikationen soll mit höchster Priorität erfolgen. Des Weiteren müssen auch bereits vorhandene Elemente der 3GPP-Systeme überarbeitet werden, um die verbesserten Funktionalitäten (z.B. IMS) zu spezifizieren.

Im Unterschied zur Arbeit des 3GGP ist für die europäischen Mobilfunknetze noch nicht klar absehbar, wann sie sich zu All-IP Netzwerken entwickeln werden. Die meisten UMTS-Netze erreichen bisher noch nicht einmal die Abdeckung des Vorgängers GSM. Da GSM immer noch der führende Mobilfunkstandard ist und UMTS-Endgeräte noch nicht in der Vielfalt wie GSM-Handys verfügbar sind, wird die Umstellung auf die neuen Spezifikationen nach dem Release '99 auch nicht mit übermäßigem Tempo vorangetrieben. Beispielsweise telefonieren nur ca. 1,4 Millionen der ca 29,2 Millionen Vodafone Kunden in Deutschland über das UMTS-Netz [ssu06]. In Japan hingegen telefonieren rund 90 Prozent der Kunden des Anbieters KDDI mit Hilfe der 3G-Technologie. HSDPA befindet sich mittlerweile bei den meisten Netzbetreibern im Testbetrieb und wird wohl im Laufe des Jahres 2006 für die kommerzielle Nutzung zur Verfügung stehen.

Insgesamt ist schwer abzuschätzen, wann genau die europäischen Netzbetreiber die Fortentwicklung zum All-IP Netzwerk umsetzten werden. Es ist aber davon auszugehen, dass etwa bis zum Jahr 2010 die neuen Funktionalitäten und Dienste zur Verfügung stehen werden und GSM endgültig von der dritten Generation abgelöst wird.

3.4.2 Fazit

Die Entwicklung zum AIPN bietet für zukünftige Mobilfunksysteme vielfältige neue Anwendungsmöglichkeiten. Durch eine einheitliche paketorientierte Plattform können für den Endkunden neue und individuell zugeschnittene Dienste verfügbar gemacht werden. Die Mobilfunknetze entwickeln sich weg von der auf Sprachdienste eingeschränkten Nutzung hin zu multimedialen Netzwerken, auf die über eine Vielzahl von Endgeräten und Zugangssystemen zugegriffen werden kann. Für die Netzbetreiber eröffnet sich ein erhebliches Ertragspotenzial, da sie neue Dienste kostengünstiger, auf Basis von verbreiteten Technologien, entwickeln können. Der Einsatz der Mobilfunknetze für verschiedene Anwendungszwecke birgt zudem Synergieeffekte und reduziert Kosten für Ausrüstung und Betrieb. Der Endanwender profitiert durch individuell zugeschnittene Dienstleistungen, garantierte Dienstgüte und erhöhte

Mobilität vom weiteren Netzausbau. Schließlich werden die heutigen Abrechnungsmodelle wie volumenabhängige Verrechnung von modernen Bezahlmodellen abgelöst, die eine Nutzung gegenüber konkurrierender Technologien wie xDSL attraktiver machen.

Um das Ziel eines All-IP Netzwerks zu erreichen, sind noch umfangreiche Weiterentwicklungen der aktuellen Infrastruktur notwendig. In erster Linie wird die Herausforderung darin bestehen, die erwartete Flut von unterschiedlichsten paketorientierten Daten zu bewältigen und die Vermittlungsnetze entsprechend auszubauen ohne dabei übermäßige Kosten entstehen zu lassen. Die weitere Zuwendung zu IP basierten Strukturen ist wohl nicht mehr aufzuhalten und wird sich allenfalls durch technische Probleme, Streitigkeiten um Standards oder wirtschaftliche Gesichtspunkte ein wenig verzögern.

Insgesamt ist die Entwicklung zum All-IP Netzwerk sowohl aus Sicht der Anwender, als auch aus Sicht der Betreiber und Investoren zu begrüßen. Die zukünftige Generation von Mobilfunknetzwerken könnte für beide Seiten völlig neue Möglichkeiten bieten und weit größere gesellschaftliche Bedeutung erlangen als ihre Vorgänger.

Literatur

[Ahre03] Peter Ahrens. *GSM Basics*. Schlembach. 2003.

[Cast04] Jonathan P. Castro. *All IP in 3G CDMA networks*. Wiley. 2004.

[Grou05] Technical Specification Group (Hrsg.). TR 22.978 V7.1.0 All-IP Network feasibility study. Studie, 3rd Generation Partnership Projekt - 3GPP, Juni 2005.

[Kroe04] Michael Kroedel. Funk Generationen - UMTS in Theorie und Praxis. *ct* (10), 2004, S. 158–160.

[Riem03] Rudolf Riemer. UMTS Grundlagen, 2003.

[ScAS05] Sibylle Schaller, Dr. Daniele Abbadessa und Dr. Anett Schuelke. Neue Dienste braucht das Land. *Funkschau* (12), 2005, S. 28–30.

[Spri02] Robert Springer, Andreas und Weigel. *UMTS*. Springer. 2002.

[ssu06] ct ssu. Vodafone D2 hat über 1,4 Millionen UMTS-Kunden, Januar 2006.

[TaBo02] Manfred Tafener und Ernst Bonek. *Wireless Internet Acess over GSM and UMTS*. Springer. 2002.

[Tane03] Andrew S. Tanenbaum. *Computernetzwerke*. Pearson. 4., überarb. Aufl.. Auflage, 2003.

Abbildungsverzeichnis

Verteiltes Schlüsselmanagement in Ad-hoc-Netzen

Abdellatif Laaroussi

Kurzfassung

Ein Mobiles Ad-hoc-Netzwerk(MANET) ist eine Menge von drahtlosen Mobilknoten, die keine Infrastruktur(Basisstation/ Access Point) und keine zentrale Verwaltung benötigen, um ein Netzwerk aufzubauen. Diese Ausarbeitung befasst sich mit verteiltem Schlüsselmanagement in Ad-hoc-Netzwerken, noch ausführlicher wird das Schwellwertverfahren dargestellt.

1 Einleitung

1.1 Motivation

In den letzten Jahren ist die Verwendung drahtloser Kommunikation sehr stark gestiegen. Es gibt Funknetze wie z.B. GSM (Global System for Mobile Communication) und UMTS (Universal Mobile Telecommunication System). Auf der anderen Seite gibt es verschiedene Standards für die mobile Vernetzung, z.B. Wireless LAN oder Bluetooth. In der letzten Zeit entwickeln Forscher eine neue Generation von drahtlosen Netzwerken, Ad-hoc-Netzwerke, die gegenüber den verwendeten Zellennetzwerken ein neues Paradigma darstellen. Ein mobiles Ad-hoc-Netzwerk ist ein selbstorganisierendes Netzwerk. Geräte (Knoten), die in Reichweite der drahtlosen Funkstelle liegen, kommunizieren direkt über die drahtlose Verbindung, während weiter entfernten Konten mittels anderer Knoten erreicht werden, die die Nachrichten als Router weiterleiten, d.h. jedes Gerät ist nicht nur eine Sende -oder Empfangstation für die Sprache oder Datenübertragung, sondern auch gleichzeitig ein Router. Mobile Ad-hoc-Netzwerke haben Vorteile gegenüber drahtlosen Infrastrukturen:

- Netze können einfach, kostengünstig und schnell aufgebaut werden,
- Robuster gegenüber dem Ausfall einzelne Komponenten.

Ad-hoc-Netzwerke werden zum größten Teil im militärischen Bereich eingesetzt: im Feld sollen über solche Netzwerke verschiedene militärische Einheiten mit taktischen Informationen versorgt und koordiniert werden. Sie werden zunehmend auch im zivilen Bereich eingesetzt, z.B. in Sensornetzwerken und in der Fahrzeugkommunikation z.B. für Taxifahrer oder Polizeistreifen. Zu den Anwendungen von Ad-hoc-Netzwerken gehören auch Rettungseinsätze in Katastrophengebieten, wenn die Infrastruktur zusammenbricht, z.B. bei Erdbeben [Schi00].

Wegen hoher Dynamik der Topologie, begrenzter Bandbreite und begrenzten Ressourcen der Endgeräte ist das Einführen eines Sicherheitssystems in Ad-hoc-Netzen sehr kompliziert.

1.2 Gliederung

Zunächst wird im Kapitel 2 gezeigt, was Sicherheitsziele und Bedrohungen sind und welche Arten von kryptographischen Verfahren es gibt. Das Kapitel 3 stellt das Schwellwertverfahren von Shamir und ein Sicherheitskonzept vor. Im Kapitel 4 gehen wir auf dass RSA-Verfahren mit dem (t,n)-Schwellwertverfahren ein. Das 5 ist eine Zusammenfassung.

2 Sicherheit

2.1 Mathematische Grundlagen

Zuerst werden ein paar Informationen über Äquivalenzklassen, Satz von Euler und über erweiterten Euklidischen Algorithmus gegeben, auf die wir später eingehen [V.W.01].

Die Äquivalenzklassen der Relation (mod n) enthalten jeweils diejenigen Zahlen, die bei Division durch n denselben Rest ergeben, sie heißen deshalb Restklassen. Die kleinste nichtnegative Zahl in jeder Restklasse heißt Repräsentant der Restklasse. Die Menge der Repräsentanten $0, 1, 2, \ldots, n-1$ wird mir Z_n bezeichnet. z.B. $Z_2 = 0, 1$.

Um ein Inverseelement in Restklassen zu berechnen, benutzt man den erweiterten Euklidischen Algorithmus:

Satz: seien $a \in N_0$, $b \in N_0$. Der grösse gemeinsame Teiler ggT(a, b) lässt sich als linearkombination von a und b darstellen.

$$ggT(a, b) = ua + vb,\ u, v \in Z.$$

$ggT(a, b) = ua + vb$ und $ggT(a, b) = 1$, d.h. $ua + vb = 1$

Modulo b gerechnet ergibt sich :

$1 \equiv ua + vb \equiv ua(\mathrm{mod}b)$ $a^{-1} \equiv u(\mathrm{mod}b)$

Damit ist $u(mod\ b)$ das Inverseelement von a in Z_b^*

In RSA-Verfahren werden wir den Satz von EULER brauchen.

Satz: Sei p eine Primzahl. Dann gilt für alle $a \in N$, die nicht durch p teilbar sind

$$a^{p-1} \equiv 1(\mathrm{mod}p)$$

Erweiterter Satz von EULER: Seien p und q Primzahlen. Dann gilt für alle $m \in N_0$ und $k \in N_0$

$$m^{k(p-1)(q-1)+1} = m(\mathrm{mod}pq) \tag{1}$$

2.2 Ziele der Sicherheit

Beispielzenario(Abbildung 1):

Alice und Bob möchten sicher kommunizieren, was bedeutet das?

Alice wünscht, dass nur Bob ihre gesendete Nachricht verstehen kann, obwohl beide über ein unsicheres Medium kommunizieren, in dem ein Eindringling (Trudy) die Nachricht der beiden abfangen, lesen und manipulieren kann. Anderseits möchte Bob sicher sein, dass die von Alice empfangene Nachricht tatsächlich von Alice stammt. Und Alice will sicher sein, dass die Person mit der sie kommuniziert, tatsächlich Bob ist. Alice und Bob möchten also sicherstellen, dass der Inhalt von Alices Nachricht auf dem Transit nicht geändert wurde. [KuRo02].

Um ein Ad-hoc Netz zu sichern, betrachten wir die folgende Punkte [KrRe00][Hauc02][ZhHa]

- Verfügbarkeit(Availability):

  Verfügbarkeit ist die ständige Präsenz von Diensten und von Ressourcen in einem Netzwerk. Diese Dienste und Ressourcen können im Optimalfall gegen Ausfälle und Angriffe jeder Art geschützt werden. Verfügbarkeit ist gewährleistet, wenn die Funktionalität von Software und Hardware nicht auf unbefugte Weise beeinträchtigt wurde.

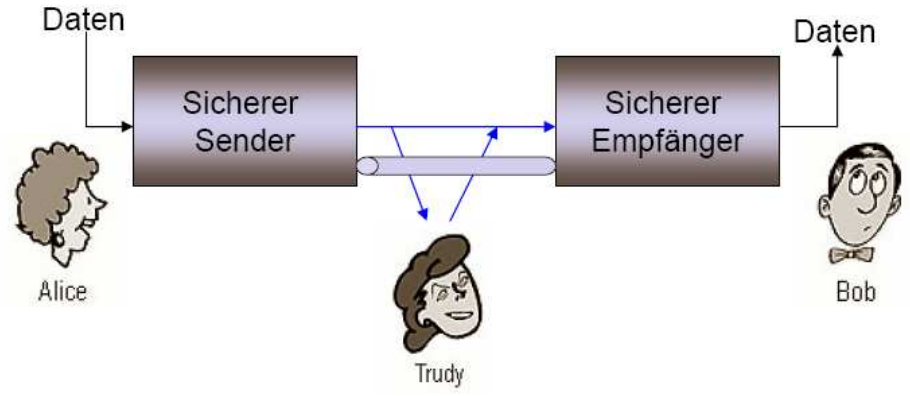

Abbildung 1: Beispiel.

- Authentizität(authenticity):

  Authentizität stellt die eindeutige Identität eines Kommunikationspartners sicher, d.h. Ein Knoten oder eine Person mit vom System überprüfter Identität bezeichnet man als authentisiert. Ohne eine solche Authentifizierung könnte sich ein Angreifer für einen anderen Knoten ausgeben und dadurch das Netzwerk stören und außerdem Zugang zu vertraulichen Daten erhalten.

- Integrität(Integrity):

  Integrität von Daten ist die Sicherstellung, dass diese Daten nicht in einer unautentisierten Art und Weise verändert werden.

- Vertraulichkeit(Confidentiality):

  Vertraulichkeit ist gegeben, wenn sichergestellt werden kann, dass Information nicht durch nicht-autorisierte Knoten oder Personen eingesehen werden können. In der Welt der elektronischen Kommunikation wird dies in der Regel mit Hilfe von Verschlüsselung realisiert.

- Nichtabstreitbarkeit:

  Wenn über ein Kommunikationssystem Transaktionen abgewickelt werden, will der Empfänger nicht nur die Identität des Absenders zweifelsfrei prüfen können (Authentizität) und sicher sein, dass die Nachricht nicht verändert wurde(Integrität), er will im Falle eines Rechtsstreits dem Absender auch zweifelsfrei nachweisen können, dass dieser die Nachricht auch wirklich gesendet hat.

2.3 Bedrohungen

Eine Bedrohung in einem Kommunikationssystem ist ein mögliches Vorkommnis, das sowohl das Werk eines böswilligen Angreifers als auch das Resultat eines an sich harmlosen Fehlers sein kann, oder jede Folge von Handlungen, welche eine oder mehrere der oben genannten Sicherheitsziele verletzt[KrRe00]. Es gibt verschiedene Arten von Bedrohungen. Der Bruch der Vertraulichkeit (disclosure threat) bezeichnet das Bekanntwerden von Informationen für Knoten oder Personen, gegenüber denen diese Informationen verborgen bleiben sollten. Unter Verletzung der Datenintegrität (integrity threat) versteht man jegliche unautorisierte Veränderung von gespeicherten oder in Übertragung befindlichen Daten. Die nicht Verfügbarkeit einen Systems (denial of service threat) bedeutet, dass eine Ressource blockiert ist und ein autorisierter Nutzer nicht darauf zugreifen kann.

Ein Angreiff ist ein nicht autorisierter Zugriffsversuch auf ein System. Der Angreifer nutzt dabei bestimmte Schwachstellen eines Systems aus, und lässt somit eine Bedrohung real werden.

Bedrohungen entstehen durch Schwachstellen. Man unterscheidet zwischen zwei Arten con Angriffen[KrRe00] :

- Passive Angriffe:

  z.B. das Medium abhören und das unbefugte Lesen von gespeicherten Daten. Passive Angriffe können meistens nicht nachgewiesen werden.

- Aktive Angriffe:

  Dazu gehören die nicht-autorisierte Verfälschungen von gespeicherten Daten, die Veränderungen der Reihenfolge, Verdoppelung oder Löschung von Datenpaketen. Aktive Angriffe können immer nachgewiesen weden.

2.4 Verschlüsselungs-Verfahren

2.4.1 Symmetrisch

Symmetrische Verschlüsselung ist ein Verfahren, bei dem Sender und Empfänger einen gemeinsamen Schlüssel benutzen, um die Nachrichten zu ver- und entschlüsseln. Sie müssen aber vorher den Schlüssel über einen sicheren Kanal austauschen. Die häufigsten symmetrischen verfahren sind Blockchiffren und Stromchifren[KrRe00].

2.4.2 Asymmetrisch

Bei asymmetrischen Verfahren werden zwei unterschiedliche Schlüssel im Ver- und Entschlüsselungsschritt benutzt. Es gibt einen öffentlichen Schlüssel e_B und einen privaten Schlüssel d_B. e_B kann einfach aus d_B berechnet werden, aber die Berechnung von d_B aus e_B ist unmöglich.

Abbildung 2zeigt die Vorgänge eines asymmetrischen Verfahrens:

Bevor Alice Bob eine Nachricht m schickt, verschafft sie sich den zu verwendenden öffentlichen Schlüssel e_B, mit dem die Nachricht m verschlüsselt wird. Die Übertragung dieses Schlüssels der Nachricht muss dann nicht mehr über einen sichern Kanal erfolgen. Bob bekommt den Chiffretext $e_b(m)$, der mit dem privaten Schlüssel d_B entschlüsselt wird. Am Ende hat Bob wieder die originale Nachricht m. Beispiele für asymmetrische Verschlüsselung sind RSA (Rivest, Shamir, Adelman), ElGamal und PGP(Pretty good Privacy) für die Verschlüsselung von E-Mails[KrRe00][Hauc02].

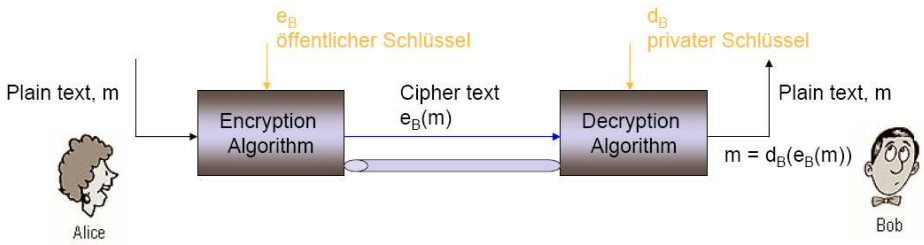

Abbildung 2: Asymmetrische Verfahren.

RSA wurde 1977 von Ron Rivest, Adi Shamir und Leonard Adelman entwickelt. Die Sicherheit von RSA liegt in der Schwierigkeit, eine sehr große Zahl in ihre Faktoren zu zerlegen.

Algorithmus(Schlüsselerzeugung):

Zunächst muss ein öffentlicher Schlüssel e und ein privater Schlüssel d erzeugt werden.

1. Man wählt zufällig zwei große Primzahlen p und q aus und berechnet $n = p * q$. Dann wird $z = (p-1)(q-1)$ berechnet.
2. Wähle eine Zahl e mit $e < z$ so, dass e und z keinen gemeinsamen Teiler hat.
3. Finde eine Zahl d so, dass $d = e^{-1} \bmod z$, d.h. $e * d - 1$ durch z dividierbar ist.

Das Tupel (e, n) bilden den öffentlichen Schlüssel und (d, n) bilden den privaten Schlüssel. Die Primzahlen p und q können vergessen werden, aber sie sollten niemals bekannt werden.

Verschlüsselung:

Um eine Nachricht m zu verschlüsseln, führt man folgende Berechnung durch: $c = m^e \bmod n$, die Nachricht m wird mit e potenziert und der Rest der Division durch n bildet den Chiffretext c.

Entschlüsselung:

Um den Chiffretext wieder zu entschlüsseln, berechnet der Empfänger: $\tilde{m} = c^d \bmod n$. Durch Potenzierung mit d lässt sich die ursprüngliche Nachricht wieder herstellen.

Jetzt wird geprüft, ob $m = \tilde{m}$

- $\tilde{m} = c^d \bmod n$
- $\tilde{m} = (m^e)^d \bmod n$
- $\tilde{m} = c^{x(p-1)(q-1)} \bmod n$
- $\tilde{m} = m \bmod n$ (Satz von EULER, siehe Abschnitt 2.1)
- $\tilde{m} = m$

Um RSA zu knacken, müsste ein Eindringling versuchen, den privaten Schlüssel d zu berechnen oder n in die Primzahlfaktoren p und q zu zerlegen. RSA kann nur solange als sicheres Verfahren gelten, wie es mit vertretbarem Aufwand praktisch nicht möglich ist eine Faktorisierung von n in p und q zu erreichen.

3 Geheimnisteilung

3.1 Defenition von Schwellwert Kryptographie (Shamir-Konzept)

Schemata zur Geheimnisteilung schützen Vertraulichkeit und Integrität von Informationen, indem sie diese Informationen auf verschiedene Orte verteilen. Die Geheimnisteilung hat für Ad-hoc-Szenarien Vorteile gegenüber den in Abschnitt 2.3 vorgestellten Authentisierungsverfahren. Zum Beispiel wird das Geheimnis nicht an einem einzigen Ort verwahrt, der gezielter Angriffspunkt sein kann oder der unerwartet ausgeschaltet bzw. unerreichbar werden kann. Eine Möglichkeit für die Realisierung der Geheimnisteilung ist die Schwellwert-Kryptographie.

Dabei geht es um das Aufteilen eines Geheimnisses, das in Form von Daten D vorliegt, auf mehrere Personen einer Gruppe. Jede einzelne Person kennt nur einen Teil von diesem Geheimnis[Hauc02][DeFr89][ZhHa] .

(t, n)-Schwellwert-Kryptographie: Ein Geheimnis wird auf eine Gruppe von n Teilnehmern verteilt, sodass eine beliebige Teilgruppe von t Teilnehmern das Geheimnis rekonstruieren kann, t-1 oder weniger Teilnehmer nicht mehr dazu in der Lage sind[A.Sh79].

Erreicht wird dies, indem auf Lagrange-Interpolation bei Polynom-Funktionen zurückgegriffen wird: für t fixe Punkte $(x_i, y_i), \ldots, (x_t, y_t)$ mit unterschiedlichen x_i gibt es genau ein Polynom $f(x)$ vom Grad $t-1$, sodass für alle i gilt $f(x_i) = y_i$; Um die geheime Information D in n Teile zu zerlegen, wird ein Polynom $f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$ von einem Verteiler gewählt $D_1 = f(x_1)$,...,$D_n = f(x_n)$ berechnet und auf n Teilnehmern verteilt. Das Polynom muss selbstverständlich geheim bleiben. Aus jeder Teilmenge der D_i mit mindestens t Elementen können nur die Koeffizienten a_i des Polynoms berechnet werden, um schließlich $D = f(0)$ zu bestimmen.Am Ende kann ein Kombinator (Combiner), der mindestens t Teile y_i von Teilnehmern $\{P_1, ..., P_n\}$ bekommen hat, das Polynom wieder herstellen.

Protokoll (Schwellwertverfahren nach Shamir)[Waet]:

Gegeben: Eine Primzahl $p \in N$, Geheimnis $k \in Z_p$, Teilnehmer $\{P_1, ..., P_n\}$ mit $n \in N$ und $p \geq n + 1$, Schwellwert $t \in N$ mit $t \leq n$.

1. Der Verteiler wählt n vierschiedene Elemente $x_i \in Z_p$, $i \in \{1, ..., n\}$.
2. Der Verteiler teilt dem Teilnehmer $P_i, i \in \{1, ..., n\}$ seinen Wert x_i mit. Alle Werte x_i sind öffentlich.
3. Der Verteiler möchte das Geheimnis k verteilen. Er wählt zufällig und geheim $t - 1$ Elemente $a_1, \cdots, a_t \in Z_p$
4. Der Verteiler bestimmt damit ein Polynom:

$$f(x) = k + \sum_{i=1}^{t-1} a_i x^i \in Z_p[x] \tag{2}$$

   von einen Grad höchstens t-1.
5. Er berechnet $y_i = f(x_i)$,$i \in \{1, ..., n\}$ und übermittelt jedem Teilnehmer $P_i, i \in \{1, ..., n\}$ auf einen sicheren Kanal seinen Teil y_i.
6. Der Kombinator(Combiner) erhält auf sicheren Wege die Teile $y_{i_1}, \cdots, y_{i_t}$ von Teilnehmern $P_{i_1}, \cdots, P_{i_t}$.
7. Der Kombinator stellt mit Hilfe der Teile $y_{i_1}, \cdots, y_{i_t}$ das Polynom wieder her.

Mit der Interpolationsformel von Lagrange wird das Polynom f bestimmt

$$f(x) = \sum_{j=1}^{t} y_{i_j} \prod_{1 \leq k \leq t, k \neq j} (x - x_{i_k}) \left(x_{i_k} - x_{i_j}\right)^{-1}. \tag{3}$$

Um das Geheimnis $k = a_0$ zu erhalten, muss man gar nicht das Polynom bestimmen, sondern es reicht, wenn der Kombinator

$$k = f(0) = \sum_{j=1}^{t} y_{i_j} \prod_{1 \leq k \leq t, k \neq j} x_{i_k} \left(x_{i_k} - x_{i_j}\right)^{-1} \tag{4}$$

$$k = f(0) = \sum_{j=1}^{t} y_{i_j} b_j \qquad \text{mit} \qquad b_j = \prod_{1 \le k \le t, k \ne j} x_{i_k} (x_{i_k} - x_{i_j})^{-1} \tag{5}$$

berechnet.

Beispiel:

- $p = 17$, $n = 5$, $k = 4$ und $t = 3$.
- Der Verteiler wählt die öffentlichen Werte: $x_1 = 1$, $x_2 = 2$, $x_3 = 9$, $x_4 = 15$ und $x_5 = 16$
- Jetzt wählt der Verteiler t - 1 = 2 zufällige Werte $a_1 = 1$ und $a_2 = 1$
- Polynom: $f(x) = 4 + x + x^2$
  - $y_1 = f(1) = 4 + 1 + 1 \bmod 17 = 6$
  - $y_2 = f(2) = 4 + 2 + 4 \bmod 17 = 10$
  - $y_3 = f(9) = 4 + 9 + 81 \bmod 17 = 9$
  - $y_4 = f(15) = 4 + 15 + 225 \bmod 17 = 6$
  - $y_5 = f(16) = 4 + 16 + 256 \bmod 17 = 4$
- und übermittelt diese den jeweiligen Teilnehmer.
- Wir nehmen an, dass P_2, P_3 und P_5 das Geheimnis wieder herstellen wollen.
- jetzt werden die b_j's berechnet...
  - $b_j = \prod_{1 \le k \le t, k \ne j} x_{i_k} (x_{i_k} - x_{i_j})^{-1} \bmod p$
  - $b_1 = x_3 (x_3 - x_2)^{-1} x_5 (x_5 - x_2)^{-1} \bmod p = 15$
  - $b_2 = x_2 (x_2 - x_3)^{-1} x_5 (x_5 - x_3)^{-1} \bmod p = 16$
  - $b_3 = x_2 (x_2 - x_5)^{-1} x_3 (x_3 - x_5)^{-1} \bmod p = 4$
- $k = b_1 * y_2 + b_2 * y_3 + b_3 * y_5 \bmod p = 310 \bmod 17 = 4$
- Am Ende hat man wieder die originale Nachricht, in dem Fall $k = 4$.

Während der gesamten Lebensdauer eines $(t-n)$- Schwellwert-Schemas ist die Geheimnisteilung gewährleistet, wenn es einem Angreifer nicht gelingt mindestens t Orte zu kompromittieren.

Verifizierbare Geheimnisteilung: Bisher musste ein Verteiler das Geheimnis zerlegen und auf die Teilnehmer verteilen. Der Verteiler stellt wieder einen zentralen Angriffspunkt dar, der für Ad-hoc-Netzwerke zu vermeiden ist. Dafür bietet es sich an, das Geheimnis von den Teilnehmern in einem Prozess gemeinsam zu rekonstruieren. So kennt keine einzelne Stelle zu keinem Zeitpunkt das komplette Geheimnis. Allerdings muss bei diesem Verfahren sichergestellt werden, dass die Teilnehmer nur korrekte Werte übermitteln und sowohl das Geheimnis als auch Teilgeheimnisse von den einzelnen Teilnehmern verifiziert werden können[Hauc02] [BHMP$^+$] .

Proaktive Geheimnisteilung: Bei der proaktiven Geheimnisteilung werden die Teile des Geheimnisses periodisch geändert, ohne dass das Geheimnis selbst geändert werden muss. Ein Angreifer hat somit nur kurze Zeit zu Verfügung, mindestens t Orte anzugreifen, so lange die Auffrischperiode gewählt wurde. Nach Ablauf einer Auffrischperiode sind alle Informationen wertlos, die ein Angreifer bisher über das Geheimnis gewonnen hat[Hauc02] [BHMP$^+$].

3.2 Sicherheitskonzept für MANETs

Ein ideales Konzept schützt eine Kommunikation gegen alle in Abschnitt 2.3 genannte Angriffe. Das Netzwerk soll offen für neue Knoten sein, d.h. es soll auch für Knoten, die dem Netzwerk bisher nicht bekannt sind, möglich sein, dem Netz beizutreten. Ein Netz von zuvor bekannten Knoten ist einfacher realisierbar, es geht jedoch die Offenheit verloren.

Ad-hoc-Netzwerke sind dynamisch, deshalb ändern sich die Vertrauensverhältnisse ständig. Es können neue Knoten in das Netz hinzukommen oder daraus verschwinden. Eine statische Konfiguration ist daher nicht möglich[Hauc02] [BHMP$^+$].

- Clusterbasiertes Ad-hoc-Netzwerk:

  Das Netz kann aus mehreren Clustern bestehen, die jeweils von einem Clusterhead (CH) organisiert werden. Im CH-Netzen gibt es auch Gateways, die Kontakte zu benachbarten Clustern herstellen. Die Clusterheads senden in bestimmten Intervallen CH-Beacons. Dies sind Nachrichten, die wichtige Informationen für Cluster-Mitglieder enthalten und als Broadcasts gesendet werden.

  CH-Beacon:

PubCH	**PubCN**	$K_1, \ldots, K_i$	$G_1, \ldots, G_j$

  CH-Beacons enthalten die öffentlichen Schlüssel vom Clusterhead (PubCH) und vom Netzwerk (PubCN), eine Liste der einzelnen Knoten und deren Status im Cluster($K_1, \cdots, K_i$) und eine Liste der Gateways ($G_1, \cdots, G_j$) im Cluster.

  Auch die Gateways senden in bestimmten Intervallen GW-Beacons. Damit können sich die Knoten ein Bild vom Netzwerk machen.

  GW-Beacon:

PubGW	$Cluster_1, \ldots, Cluster_n$	$StatusinCluster_i$

  GW-Beacons bestehen aus dem öffentlichen Schlüssel des Gateways (PubGW), Informationen über ihre benachbarten Cluster ($Cluster_1, \cdots, Cluster_n$) und aus dem Status, den diese in den jeweiligen Clustern(Status in Cluster i) haben.

  Ein CH-Netzwerk, das die einzelnen Clusterheads zusammen bilden, besitzt ein eigenes Privat-Key-Schlüsselpaar. Der private Netzschlüssel wird im CH-Netzwerk mittels proaktiver Geheimnisteilung verteilt. Der öffentliche ist jedem Knoten bekannt, da er in den CH-Beacons enthalten ist. Dazu unabhängig hat jeder Knoten ein eigenes PK-Schlüsselpaar für die sichere Kommunikation.

  Die Teilgeheimnisse müssen ständig neu auf die neuen Mitglieder verteilt werden, wenn die Zusammensetzung des CH-Netzwerk sich verändert. Das passiert immer, wenn ein Knoten in CH-Netzwerk ein- oder austritt. Dieser Vorgang kann mit der Schlüsselauffrischung kombiniert werden, d.h. wird die Struktur des CH-Netzwerkes verändert, ist auch ein idealer Zeitpunkt die Teilgeheimnisse aufzufrischen.

- Anmeldeprozedur:

  Wird ein Knoten in einem Cluster aufgenommen, so hat er zuerst den Status eines Gastknotens ohne jegliche Rechte. Erst wenn der öffentliche Schlüssel des Knotens vom CH-Netz signiert wurde, ist der Knoten volles Cluster-Mitglied und kann Rechte über sogenannte Autorisierungszertifikate zugewiesen bekommen. Kommt ein neuer Knoten A in das Netz, wartet er auf CH-Beacons. Empfängt er CH-Beacons, so sendet er seine Bewerbung an den verantwortlichen CH. Empfängt A keine CH-Beacons, bildet er einen eigenen Cluster und ernennt sich selbst zum CH. Er generiert einen geheimen symmetrischen Cluster-Schlüssel und sendet selbst CH-Beacons.

- Authentisierung :

  Es genügt im Allgemeinen nicht, dass sich ein Knoten am Netz authentisiert; das Netz muss sich auch beim Knoten authentisieren. Ein Angreifer könnte sonst ein anderes Netz simulieren und so eventuell versuchen, bei Anmeldeversuchen von Knoten Geheimnisse zu erkunden. In dem vorgestellten Sicherheitskonzept müssen die Knoten sich gegenseitig vertrauen und für ihre Vertrauensaussagen bürgen. Ein Knoten kann seinen Schlüssel erst zertifizieren lassen, wenn er mindestens einen Bürgen für sich gefunden hat, der für seine Identität bürgt. Mit dieser Bürgschaft kann sich der Knoten beim CH-Netzwerk für eine Signatur anmelden. Abbildung 3 zeigt die Vorgänge einer erfolgreichen Authentisierung.

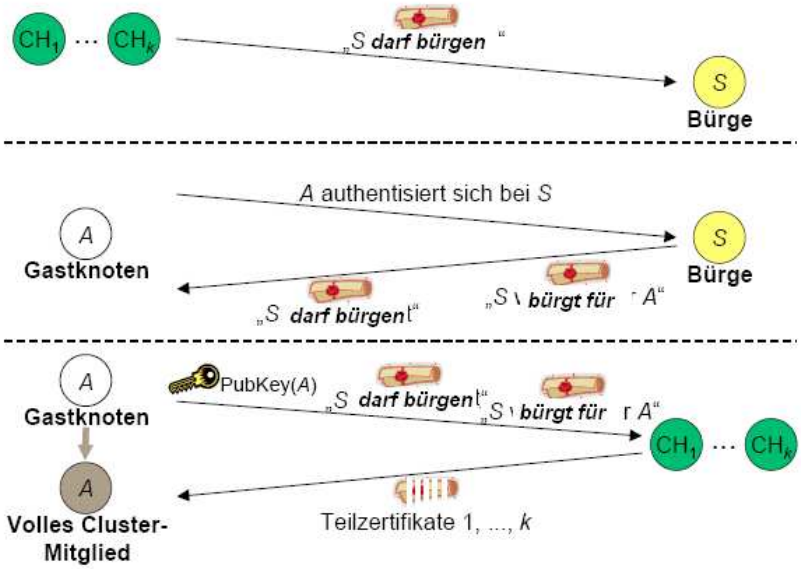

Abbildung 3: Authentisierung.

Ein Bürge wird nur dann für einen Knoten bürgen, wenn ihm dessen wahre Identität bekannt ist. Dies kann mittels physikalischen Kontakts oder über einen sicheren Kanal laufen. Je mehr Bürgen ein Knoten für sich finden kann, desto sicherer ist seine Identität.

BürgZert-Zertifikate sind diejenigen Zertifikate, die ein Bürge B einem neuen Knoten A ausstellt, um für ihn zu bürgen.

BürgZert:	**KnotenA**	**PubA**	**Dauer** t	**ich bürge**	**SigB**

BürgZert enthält die Kennung des Knotens A (Knoten A), den öffentlichen Schlüssel des Knotens (PubA), einen Zeitraum (Dauer t), für den das Zertifikat gültig ist, die Kennung (ich bürge), wofür das Zertifikat gilt und ein Signatur des Bürgen (SigB), um seine Echtheit zu gewährleisten.

Bevor die CHs ihre Teilzertifikate an den jeweiligen Knoten senden, müssen sie erst sicherstellen, dass der Bürge auch dazu autorisiert ist. Dies geschieht durch ein Autorisierungszertifikat BürgAutZert, das der Bürge von CH-Netzwerk bekommt. Von diesem Zertifikat sendet der Bürge eine Kopie an A. Mit beiden Zertifikaten BürgZert und BürgAutZert kann sich A beim CH-Netzwerk bewerben und seinen Schlüssel zertifizieren lassen.

BürgAutZert:	**BürgeB**	**PubB**	**darf bürgen**	**SigNetz**

BürgAutZert enthält die Kennung des Knotens B(Bürge B), den öffentlichen Schlüssel (PubB) und die Information (darf bürgen), wofür der Knoten berechtigt ist. Dieses Zertifikat wird vom CH signiert(SigNetz).

Die CHs überprüfen ihrerseits das vom bewerbenden Knoten A gesendete BürgAutZert und senden A ihren Teil des Identifikationszertifikats IDZert. Erhält A genügend Teilzertifikate, so kann er das vollständige IDZert zusammensetzten.

IDZert:	**KnotenA**	**PubA**	**Dauer** t	**SigNetz**

IDZert enthält die Kennung des Knotens A (Knoten A), den öffentlichen Schlüssel des Knotens A (PubA) und einen Zeitraum (Dauer t), für den es gültig ist. Dieses Zertifikat wird auch vom CH signiert(SigNetz).

Ist ein Knoten im Besitz eins IDZert-Zertifikates, ist er nicht mehr Gastknoten, sondern ist damit zum vollwertigen Cluster-Mitglied geworden.

4 RSA-TC in MANET

Im folgenden Abschnitt werden wir über das RSA-Verfahren mit dem (t,n)-Schwellwertverfahren von Shamir reden. Man wählt einfach zwei grosse Primzahlen p und q mit $p = 2\acute{p}+1$ und $q = 2\acute{q}+1$ ($\acute{p}$ und $\acute{q}$ sind auch Primzahlen) und daraus folgt $m = pq$ [ErCh] [Waet].

Dann wählt man ein $\lambda(m) = \acute{p} * \acute{q}$ ($\lambda(m)$ wird auch als Carmichael-Zahl von m bezeichnet) und eine öffentliche Zahl e mit $1 < e < \lambda(m)$ und $ggT(e, \lambda(m)) = 1$, d.h. e und $\lambda(m)$ haben keinen gemeinsamen Teiler.

Aus e bestimmt man eine Zahl $d = e^{-1} \bmod \lambda(m)$, d.h. $ed = x\lambda(m) + 1$ für ein geeignetes $x \in N$. Wie im Abschnitt 3.1 hat man ein Geheimnis k, das er auf n Teilnehmer verteilen will, sodass t von ihnen das Geheimnis k zusammen rekonstruieren können.

Verfahren von (t,n)-RSA-Schwellwertverfahren

Gegeben: Geheimnis $k \in Zp$, Teilnehmer $P_1, \cdots, P_n$, $n \in N$ und Schwellwert $t \in N$ mit $t \leq n$. t und n sind öffentlich

Verschlüsselung:

- Der Verteiler wählt zwei grosse Primzahlen $p = 2\acute{p} + 1$ und $q = 2\acute{q} + 1$ und berechnet daraus $m = p * q$. Dann wählt man e mit $1 < e < \lambda(m)$ und $ggT(e, \lambda(m)) = 1$ Er berechnet $d = e^{-1} \bmod \lambda(m)$ und veröffentlicht (m, e), d bleibt aber geheim.
- Jetzt verwendet der Verteiler das Schwellwertverfahren (s. Abschnitt 3.1) Er wählt n verschiedene ungerade Elemente $x_i \in Z_{\lambda(m)}$, $i \in \{1, \cdots, n\}$ die Differenzen von x_i's sind alle modulo $\acute{p}$ und $\acute{q}$ verschieden von 0, er bestimmt auch ein Polynom f des Grades höchstens $t - 1$ über $Z_{\lambda(m)}$, sodass alle f(x_i) gerade sind und $f(-1) = d - 1$ gilt.
- Der Verteiler berechnet

$$y_i = f(x_i)(\prod_{1 \leq j \leq n, j \neq i} (x_i - x_j)^{-1}) \bmod \acute{p}\acute{q}. \tag{6}$$

und übermittelt jedem Teilnehmer P_i auf einem sicheren Kanal seinen Teil y_i, $i \in \{1, \cdots, n\}$

- Jetzt berechnet der Verteiler $C = M^e \bmod m$ und übermittel diese Nachricht eine Gruppe der Teilnehmer

Entschlüsselung:

- Jeder Teilnehmer $P_{i_j}, j \in \{1, \cdots, t\}$, berechnet

$$c_{i_j} = C^{y_{i_j}} \bmod m \tag{7}$$

- Der Zusammensetzer erhält auf sicherem Wege die Werte c_{i_j} und berechnet

$$\hat{c}_{i_j} = c_{i_j}^{\prod_{1 \le i \le n, i \notin [i_1, \cdots, i_t]} (x_{i_j} - x_i) \prod_{1 \le k \le t, t \ne k} (-1 - x_{i_k})} \bmod m. \tag{8}$$

- Er erhält die Nachricht M durch

$$M = C \cdot \prod_{j=1}^{t} \hat{c}_{i_j} \bmod m \tag{9}$$

Beispiel:

- Der Verteiler wählt die Primzahlen $p = 2 * 5 + 1 = 11$ und $q = 2 * 11 + 1 = 23$.
- Dann ist $m = 253, \acute{p} = 5, \acute{q} = 11$ und $\lambda(253) = 110$.
- Er wählt $e = 3$ und berechnet dazu $d = 3^{-1} \bmod 110 = 37$.
- (253,3) ist der öffentliche Schlüssel und $d = 37$ ist der geheime.
- Der Verteiler wählt die öffentlichen Werte: $x_1 = 1, x_2 = 3, x_3 = 5$ und $x_4 = 7$ deren Differenzen gerade und modulo 5 und 11 verschieden von 0 sind.
- Er wählt $a_0 = 10, a_1 = 3$ und $a_2 = 29$.
- Polynom: $f(x) = 10 + 3x + 29x^2 \in Z_{110}$
- Dann berechnet er die y_i´s
  - $y_1 = f(x_1)((x_1 - x_2)(x_1 - x_3)(x_1 - x_4))^{-1} \bmod 55 = 6$
  - $y_2 = f(x_2)((x_2 - x_1)(x_2 - x_3)(x_2 - x_4))^{-1} \bmod 55 = 45$
  - $y_3 = f(x_3)((x_3 - x_1)(x_3 - x_2)(x_3 - x_4))^{-1} \bmod 55 = 15$
  - $y_4 = f(x_4)((x_4 - x_1)(x_4 - x_2)(x_4 - x_3))^{-1} \bmod 55 = 44$
- Knoten A möchte die Nachricht $M = 211$ chiffrieren.
- Er berechnet $C = 211^3 \bmod 253 = 41$
- und sendet diesen Wert an Knoten B und an die Teilnehmer P_1, P_2, P_4.
- Jeder dieser Teilnehmer bestimmt $c_i = C^{y_i} \bmod m, i \in 1, 2, 4$,
  - $c_1 = 41^6 \bmod 253 = 146$,
  - $c_2 = 41^{45} \bmod 253 = 87$,
  - $c_4 = 41^{44} \bmod 253 = 70$,

- und sendet diesen Werten an Knoten B.
- Knoten B berechnet dann die Werte $\hat{c}_i$´s
  - $\hat{c}_1 = 146^{-4(-4)(-8)} \bmod 253 = ((((146)^4)^4)^8)^{-1} \bmod 253 = 163$,
  - $\hat{c}_2 = 87^{-2(-2)(-8)} \bmod 253 = 133$,
  - $\hat{c}_4 = 70^{2(-2)(-4)} \bmod 253 = 70$.
- Am Ende berechnet er dann $M = C \cdot \hat{c_1} \cdot \hat{c_2} \cdot \hat{c_4} \bmod m = 41 \cdot 163 \cdot 133 \cdot 70 \bmod 253 = 211$.

5 Fazit

Die Sicherheit des RSA-Schellwertverfahren beruht auf dem Problem, große Zahlen in ihre Faktoren zu zerlegen. Es ist praktisch unmöglich auf die Zahlen p und q zu kommen. Jedoch hat RSA-Schellwertverfahren auch einige Nachteile, die es schwierig machen, es in MANET einzusetzen:

- Es gibt nicht immer die Inverse für alle Zahlen in (p-1)(q-1).
- Wegen der exponentiellen Berechnungen benötigt ein RSA-Schellwertverfahren eine große Berechnungskapazität, Bandbreite und Energie.

ECC(Elliptic Curve Cryptography)-Schellwertverfahren konnte eine bessere Wahl in MANET sein. ECC-Schellwertverfahren hat gegenüber anderen Verfahren einen entscheidenden Vorteile [Hauc02]:

- eine geringere Schlüsselgröße führt zu mindestens gleichwertiger Sicherheit
- In den letzten Jahren werden ECC-Verfahren in immer mehr Bereichen eingesetzt z.B.: Navigationssystemen, Radarfallen

Anhand der Mathematik ist deutlich zu erkennen, das dieses Verfahren wesentlich komplizierter ist als andere asymmetrische Verfahren.

Literatur

[A.Sh79] A.Shamir. *How to share a secret.* Communumication ACM. 1979.

[BHMP+] Marc Bechler, Achim Hauck, Daniel Mueller, Frank Paehlke und Lars Wolf. Ein Sicherheitskonzept für clusterbasierte Ad hoc Netze.

[DeFr89] Y. Desmedt und Y. Frankel. Threshold cryptosystems. Technischer Bericht, 1989.

[ErCh] Levent Ertaul und Nitu Chavan. Security of Ad Hoc Networks and Threshold Cryptographie.

[Hauc02] Achim Hauck. Sicherheitskonzept für Ad-hoc-Netze. 2002.

[KrRe00] Gerhard Krüger und Dietrich Reschke. *Lehr und Übungsbuch Telematik.* Muenchen, Wien. 2000.

[KuRo02] James F. Kurse und Keith W. Ross. *Computernetze.* Addison Wesley, München. 2002.

[Schi00] Jochen Schiller. *Mobilkommunikation.* Techniken für das allgegenwärtigen Internet. 2000.

[V.W.01] V.Drumm und W.Weil. *Lineare Algebra und Analytische Geometrie.* Karlsruhe. 2001.

[Waet] Dietmar Waetjen. Secret Sharing und gruppenorientierte Kryptographie.

[ZhHa] Lidong Zhou und Zygmunt J. Haas. Securing Ad Hoc Networks.

Abbildungsverzeichnis

Secure OLSR - Angriffszenarien und Schutzmechanismen

Björn Hahnenkamp

Kurzfassung

Das „Optimized Link State Routing Protocol“ (OLSR, [ClJa03]) wird in Mobilen Ad Hoc Netzen (MANETs) eingesetzt. Es zählt zu den proaktiven und tabellenbasierten Routingprotokollen. Per se bietet OLSR keinerlei Sicherheitsmechanismen und ist daher nur bedingt einsetzbar. Ein Überblick sowohl über Angriffszenarien auf Routingprotokolle im Allgemeinen als auch OLSR im Speziellen wird gegeben. Weiterhin werden verschiedene Indizien für Angriffe vorgestellt. Um diesen Angriffen zu begegnen wurde Secure OLSR [ACJL+03, AdRM04] entwickelt, das ebenfalls vorgestellt wird. Am Ende dieser Arbeit steht eine Bewertung, inwiefern durch Secure OLSR Angriffe auf OLSR verhindert werden können sowie ein Ausblick auf weitere zu lösende Probleme.

1 Einleitung

Ein Mobiles Ad Hoc Netz (MANET) stellt besondere Anforderungen an das Routing. Eine hierarchische oder statische Struktur der Verbindungen auf Schicht 2 ist nicht gegeben. Das Routingverfahren muss Änderungen in der Struktur des Netzes berücksichtigen. Durch den Einsatz von drahtlosen Technologien und die dadurch erreichte Mobilität der Knoten kann sich die Menge der Nachbarknoten ändern, mit denen ein gegebener mobiler Knoten kommunizieren kann. Das Optimized Link State Routing Protocol (OLSR), beschrieben im RFC 3626 [ClJa03], beschreibt ein Routingverfahren, welches diesen Anforderungen genügt.

Wegen der Nutzung von drahtlosen Verfahren auf Schicht 2 sind MANETs leicht abhör- und angreifbar. OLSR bietet keinen Schutz gegen Angriffe auf das Routingprotokoll. Dies wird auch im Draft der IETF zu OLSR Version 2 [Clau05] explizit erwähnt. Im Folgenden sollen mögliche Angriffe auf OLSR, Möglichkeiten der Entdeckung von Angriffen und entsprechende Gegenmaßnahmen vorgestellt werden. Kein Bestandteil dieser Arbeit sind Angriffe auf anderen Schichten, zum Beispiel durch Jamming auf der physikalischen Schicht. Ebensowenig werden Angriffe auf den Nutzdatenverkehr von Netzen, die OLSR als Routingprotokoll nutzen, behandelt.

In Abschnitt 2 werden zu Beginn Grundlagen erläutert. Die Funktionsweise von OLSR wird erklärt, außerdem werden die in den nachfolgenden Teilen verwendeten Begriffe zur Netzsicherheit definiert. In Abschnitt 3 werden zunächst allgemeine Angriffe auf Routingprotokolle in MANETs aufgezeigt. Weiterhin wird auf spezielle Angriffe auf OLSR eingegangen und es werden Indizien vorgestellt, die den Knoten zur Erkennung von Angriffen dienen können. Abschnitt 4 befasst sich mit Secure OLSR als einem Ansatz, verschiedenen Angriffen auf OLSR zu begegnen. Hierbei wird ebenfalls analysiert, welche der vorgestellten Angriffe auf OLSR durch Secure OLSR nunmehr vereitelt werden. Ein knapper Überblick über weitere Arbeiten zum Thema wird in Abschnitt 5 gegeben. In Abschnitt 6 werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick gegeben. In Abschnitt 3 werden zunächst allgemeine Angriffe auf Routingprotokolle in MANETs aufgezeigt. Weiterhin wird auf spezielle Angriffe auf OLSR eingegangen und es werden Indizien vorgestellt, die den Knoten zur Erkennung

von Angriffen dienen können. Abschnitt 4 befasst sich mit Secure OLSR als einem Ansatz, verschiedenen Angriffen auf OLSR zu begegnen. Hierbei wird ebenfalls analysiert, welche der vorgestellten Angriffe auf OLSR durch Secure OLSR nunmehr vereitelt werden. Ein knapper Überblick über weitere Arbeiten zum Thema wird in Abschnitt 5 gegeben. In Abschnitt 6 werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick gegeben.

2 Grundlagen

2.1 Grundlegende Funktionsweise von OLSR

OLSR wurde im RFC 3626 der IETF [ClJa03] standardisiert, wobei auch bereits ein Draft für eine zweite Version [Clau05] existiert.

OLSR ist ein proaktives, tabellenbasiertes Routingverfahren. Bei proaktiven Routingverfahren tauschen die beteiligten Knoten ständig Topologieinformationen aus, auch wenn keine Datenübertragung stattfindet. Jeder Knoten kennt also stets die Topologie des Netzes. Bei Bedarf kann der Knoten die Route für anstehende Nutzdaten sehr schnell berechnen, da das benötigte Wissen hierfür bereits lokal verfügbar ist. Tabellenbasierte Routingverfahren berechnen überdies hinaus stets die Routen zu anderen Knoten und speichern sie in einer Tabelle. Steht eine Datenübertragung an, so kann die Route sehr schnell aus dieser lokalen Tabelle abgelesen werden.

Der Nachteil von proaktiven Verfahren ist der Overhead: Nicht jede ausgetauschte Topologieinformation wird benötigt, was dadurch wird das Medium unnötig belegt. Die Berechnung der Routingtabellen kostet Rechenzeit, und das lokale Topologiewissen erfordert Speicherplatz im mobilen Knoten. Reaktive Routingverfahren berechnen hingegen Routen erst bei Bedarf. Diese Verfahren bieten sich insbesondere dann an, wenn Knoten nur selten und nur zu wenigen anderen Knoten Daten senden und diese Übertragung nicht zeitkritisch ist.

OLSR definiert verschiedene Nachrichtentypen, mit denen Knoten Informationen austauschen. Nachrichten werden über OLSR-Pakete ausgetauscht, die über UDP (Port 698) versendet werden. Pakete werden mit einer Sequenznummer und einer Längenangabe versehen und können jeweils mehrere Nachrichten unterschiedlichen Typs beinhalten. Jede Nachricht wird unter anderem mit einer Sequenznummer (message sequence number, MSN) und der Absenderadresse gekennzeichnet. Wissen über die Topologie wird über zwei wesentliche Nachrichtentypen ausgetauscht: HELLO und Topology Control (TC).

HELLO-Nachrichten werden nur zwischen unmittelbaren Nachbarn ausgetauscht, sie werden nie weitergeleitet. Mit ihnen tauschen Nachbarn ihr Wissen über ihre jeweilige 1-hop Nachbarschaft aus, also die Menge der direkt erreichbaren Knoten. Dadurch kennt jeder Knoten seine 2-hop Nachbarschaft.

Aus seinen 1-hop Nachbarn wählt jeder Knoten eine bestimmte, möglichst kleine Teilmenge aus, die sogenannten Multipoint Relays (MPR). Diese werden so gewählt, dass jeder 2-hop-Nachbar von mindestens einem MPR direkt erreichbar ist. Die Wahl der MPRs wird ebenfalls über HELLO-Nachrichten an die Nachbarn signalisiert. MPRs speichern die Menge der Knoten, von denen sie zum MPR gewählt wurden (MPR selectors). Durch das Wahlverfahren der MPRs wird sichergestellt, dass das verteilte Wissen der MPRs bereits die globale Netztopologie umfasst.

Über TC-Nachrichten tauschen die MPRs Topologieinformationen untereinander aus und geben dieses Wissen an ihre MPR selectors weiter. So gibt jeder MPR die Information, welche Knoten er direkt erreichen kann, an alle Knoten im Netz bekannt. Jeder Knoten hat dadurch

genug Wissen über die Topologie des ganzen Netzes, um Routen zu allen Knoten zu berechnen. TC-Nachrichten werden nur von MPRs weitergeleitet. Alte TC-Nachrichten werden anhand einer speziellen Sequenznummer, der Advertised Neighbor Sequence Number (ANSN) erkannt und verworfen. Diese Art der Verteilung des globalen Topologiewissens ist effizienter als normales Fluten, wodurch OLSR mit wenig Overhead auskommt.

2.2 Weitere Elemente des OLSR-Standards

OLSR kann mehrere Interfaces pro Knoten nutzen. Informationen über solche Knoten werden mit Hilfe von Nachrichten des Typs Multiple Interface Declaration (MID) verteilt. Knoten, die Verbindungen zu nicht-OLSR Netzen haben, können dies mit Hilfe von Nachrichten des Typs Host and Network Association Nachrichten (HNA) bekannt geben.

Der OLSR Standard gibt nicht nur vor, wie Nachrichten weiterzuleiten sind, sondern auch, wie Nachrichten in den Knoten verarbeitet werden müssen. Beispielalgorithmen für die Berechnung der Routingtabelle sowie zur geeigneten Wahl der MPRs werden angegeben, andere Algorithmen können jedoch verwendet werden. Es wird festgelegt, welche Datenbasis jeder Knoten aus den Nachrichten ableiten und verwalten muss. Dazu gehören:

- Link Set: Die Menge der lokal verfügbaren Links, die zur Datenübertragung genutzt werden können
- Menge der unmittelbaren Nachbarn und Menge der 2-hop Nachbarn
- MPR Set und MPR Selector Set: Gewählte MPRs und jene Knoten, die den Knoten zum MPR gewählt haben
- Topology Information Base: globale Topologie, als Grundlage für die Berechnung der Routingtabellen
- Multiple Interface Association Information Base: Information über Knoten mit mehreren Interfaces

Neben diesen Kernfunktionalitäten, die jeder Knoten unterstützen muss, kann OLSR um weitere Funktionalität erweitert werden. Informationen des Schicht-2 Protokolls können genutzt werden, um effektiver an Informationen über die Topologie zu gelangen. Redundanz kann genutzt werden, um das Routing robuster zu machen.

2.3 Elementare Sicherheitsbegriffe

Folgende Begriffe von Sicherheit sind im Zusammenhang mit Routingverfahren wichtig:

Geheimhaltung: Die übertragene Information ist nur berechtigten Empfängern zugänglich. In bestimmten Fällen sollen Routinginformationen geheim gehalten werden (z.B. Ortsangaben bei militärischen Anwendungen). Eine Manipulation des Routings dahingehend, dass ein Teil des Datenverkehrs stets über bestimmte Knoten geroutet wird (sinkhole Angriff), ermöglicht Angriffe auf den Nutzdatenstrom.

Integrität: Die übertragene Nachricht wurde nicht manipuliert. Die Integrität der Routinginformationen muss gewährleistet sein, damit das Routing funktioniert. In Bezug auf die Nutzdaten ist das Routing ebenfalls kritisch, wenn Datenverkehr als Folge eines Angriffs über bestimmte Knoten geroutet wird, die mit Hilfe weiterer Angriffe die Nutzdaten manipulieren können.

Nicht-Abstreitbarkeit: Der Absender hat eine Nachricht wirklich gesendet.

Authentifizierung: Der Absender ist derjenige, für den er sich ausgibt. Zahlreiche Angriffe auf Routingverfahren basieren darauf, dass Nachrichten mit falschen Absenderkennungen geschickt werden, z.B. alle Verfahren zum Identity Spoofing.

Zugriffskontrolle: Berechtigung, auf ein Medium zu schreiben oder davon zu lesen.

Verfügbarkeit: Dienst kann ordnungsgemäß durchgeführt werden. Die meisten Angriffe auf das Routing haben die Isolation von Knoten, den Wegfall von bestimmten Verbindungen oder die Reduktion von Bandbreite zum Ziel. Damit wird sowohl die Funktionalität des Routings als auch die Verfügbarkeit von Nutzdatenverbindungen eingeschränkt.

3 Angriffszenarien

In diesem Abschnitt werden Angriffe auf Routingverfahren vorgestellt. In Abschnitt 3.1 wird beschrieben, welche Ziele ein Angriff auf ein Routingverfahren haben kann. Abschnitt 3.2 behandelt allgemeine Formen von Angriffen auf Routingprotokolle in MANETs, Abschnitt 3.3 geht auf konkrete Angriffe auf OLSR-Netze ein. In Abschnitt 3.4 geht es um Indizien für Angriffe, anhand derer Angriffe erkannt werden können.

3.1 Sicherheit von Routingverfahren

Angriffe auf Routingverfahren können verschiedene Ziele haben (vgl. dazu die Definitionen in Abschnitt 2.3). Ein Angreifer kann dafür sorgen, dass er den Verkehr von bestimmten oder allen Knoten abhören, manipulieren oder unterbinden kann. Hierzu wird er die im Netz bekannten Topologieinformationen so verfälschen, dass die Routenberechnung auf einer falschen Datenbasis beruht. Wird Verkehr über einen Angreifer geroutet, so kann dieser die Verbindung beliebig kontrollieren. In bestimmten Situationen ermöglicht erst eine solche Kontrolle einen Angriff auf Protokolle in höheren Schichten.

Die Verfügbarkeit des Netzes kann insgesamt gestört werden, so das Nutzdaten nicht mehr zuverlässig übertragen werden. Ein Angriff kann auch eine Reduktion der verfügbaren Bandbreite zur Folge haben. Dies kann zum Beispiel geschehen, indem indem eine effiziente Routenberechnung verhindert oder der Overhead des Routingprotokolls maximiert wird.

Die Verfügbarkeit des Netzes kann für einzelne Knoten teilweise oder vollständig eingeschränkt werden (Isolation). Außerdem können Teile des Netzes abgespaltet werden (Partition), die Knoten der einzelnen Bereiche können nach wie vor untereinander kommunizieren. Kritisch sind zum Beispiel Angriffe, bei denen Knoten mit Verbindungen zu anderen Netzen isoliert oder abgespaltet werden.

3.2 Angriffe auf Routingverfahren in MANETs

Angriffe auf das Routing in MANETs basieren darauf, dass sich ein Knoten nicht standardkonform verhält. Dieses Fehlverhalten lässt sich in zwei wesentliche Kategorien einteilen: Generierung fehlerhafter Nachrichten oder inkorrekte Weiterleitung von Nachrichten. Im ersten Fall kann weiterhin unterschieden werden, ob der Knoten eine falsche Identität angibt (Identity Spoofing) oder eine falsche Topologieinformation verbreitet (Link Spoofing).

Folgende Klassen von Angriffen auf Routing in MANETs sind bekannt und sollen hier nur kurz vorgestellt werden. Kombinationen von derartigen Angriffen sind durchaus möglich. Eine ausführlichere Beschreibung findet sich z.B. in [Raff05]:

Cache Poisoning: Dabei gibt ein Knoten fälschlich an, Verbindung zu einem (entfernten) Knoten zu haben. Knoten berücksichtigen diesen nicht existierenden Link beim Aufbau der Routingtabellen, die dadurch fehlerhaft sind. Berücksichtigt ein Knoten diesen Link bei der Routenberechnung, so kann der Angreifer den Datenstrom zu diesem Knoten beliebig kontrollieren.

Denial Of Service Angriffe: Beim *Message Bombing* wird eine Flut von Nachrichten gesendet, die Bandbreite und Rechenkapazität bei den anderen Knoten beanspruchen. Diese Ressourcen fehlen für die sonstige Abwicklung des Protokolls. Der *Shrew Angriff* zielt auf das TCP Protokoll: Durch Message Bombing mit Routing-Nachrichten wird eine TCP-Verbindung gestört. Durch die gezielte Wiederholung des Angriffes zu den Zeitpunkten, zu denen TCP eine Neuübertragung probiert, kann ein Verbindungsabbruch erzielt werden. Der *Jellyfish Angriff* bringt mit verschiedenen Methoden ebenfalls durch besonderes Verhalten des Routings die TCP-Staukontrolle zu Verbindungsabbrüchen.

Blackhole Angriffe: Mit „schwarzen Löchern“ sind Knoten gemeint, die bestimmte oder alle Pakete verwerfen, die zur Weiterleitung bestimmt sind.

Wormhole Angriffe: Ein Angreiferknoten leitet Nachrichten transparent weiter (bzw. mehrere Angreifer tunneln sie durch das Netz), und täuscht damit eine tatsächlich nicht existierende Verbindung vor. Diese Verbindung kann der Angreifer beliebig kontrollieren, zum Beispiel um Pakete zu verwerfen (Blackhole). Der Untschied zum Cache Poisoning besteht darin, dass ein Teil der Nachrichten tatsächlich zum angegriffenen Knoten weitergeleitet wird.

Manipulation von Nachrichten: Diese Klasse von Angriffen bezeichnet Verfahren, bei denen Nachrichten vor der Weiterleitung verändert werden.

Replay Angriffe: Dies beinhaltet die Speicherung von gültigen (zum Beispiel signierten) Nachrichten und deren Wiedergabe zu einem späteren Zeitpunkt. Dadurch werden alte Informationen für das Routing benutzt, was dadurch gestört wird.

3.3 Angriffe auf OLSR

OLSR bietet per se keine Sicherheit des Routings gegenüber gezielten Angriffen, dies ist im entsprechenden Standard explizit erwähnt. Im Folgenden sollen Angriffe vorgestellt werden, bei denen (mit Ausnahme des/der Angreifer/s) die Knoten ein Routing gemäß des Standards implementieren. Einen guten Überblick über diese Angriffe und deren Auswirkungen wird in [Raff05] gegeben. Eine weitere Analyse einiger aufgeführter Angriffe findet sich in [AdRM04].

3.3.1 Identity Spoofing mit Generierung falscher HELLO-Nachrichten

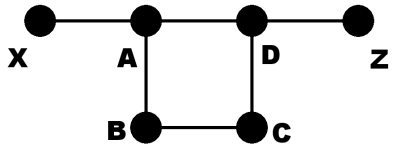

Abbildung 1: Knoten X wählt A als MPR und gibt dabei Z als Absenderkennung an

Dieser Angriff basiert auf Identity Spoofing. Ein angreifender Knoten X sendet HELLO-Nachrichten mit der Absenderkennung des Knotens Z ab, der Ziel des Angriffes ist. Die

Nachbarknoten von X gehen davon aus, dass sie Z direkt erreichen können, was nicht der Fall ist, und machen diese Information im Netz bekannt. X wählt überdies hinaus mit der Absenderkennung von Z einen seiner Nachbarn als MPR. Pakete an Z werden über die von X gewählten MPRs geroutet, und erreichen Z daher nicht. Im Beispiel (vgl. Abbildung 1) wird Knoten A zum vermeintlichen MPR von Z, wodurch B keine Pakete an Z senden kann.

3.3.2 Identity Spoofing mit falscher WILLINGNESS-Information

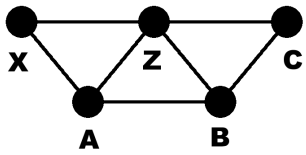

Abbildung 2: Knoten X verhindert die Wahl von A, B und C zum MPR von Z

Hierbei generiert ein angreifender Knoten X HELLO-Nachrichten im Namen der Nachbarknoten des Zieles Z (vgl. Abbildung 2). Die generierten Pakete beinhalten die vermeintliche Angabe der Nachbarn von Z, nicht MPR werden zu wollen. Hierzu wird in den HELLO-Nachrichten die Option WILLINGNESS auf den Wert WILL NEVER gesetzt. Z wird nun keinen seiner Nachbarn als MPR wählen. Dadurch kann es passieren, dass Z keine Verbindung zu einem MPR hat und somit vom Netz isoliert wird. Ebenso kann X erzwingen, dass er MPR von Z wird. Dadurch kann X allen eingehenden Verkehr für Z kontrollieren.

3.3.3 Link Spoofing durch Angabe falscher Nachbarschaften

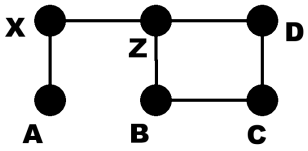

Abbildung 3: Knoten X gibt Nachbarschaft zu C an und wird somit einziger MPR von Z

Durch HELLO-Nachrichten gibt ein Knoten an, welche Knoten in seiner Nachbarschaft liegen. Ein angreifender Knoten X kann hierbei Knoten angeben, zu denen er tatsächlich keinen Link hat. Im Beispiel (siehe Abbildung 3) ist dies Knoten C. Die Nachbarn von X berechnen dadurch eine falsche 2-hop Nachbarschaft, und wählen als Folge ein inkorrektes Set von MPRs. Wird außer dem Angreifer keiner der tatsächlichen Nachbarn von Z zum MPR gewählt, so kann der Angreifer den Datenstrom zu Z beliebig kontrollieren.

Ein ähnlicher Effekt entsteht, wenn X nicht alle seine Nachbarn bekannt gibt. Dies kann ebenso dazu führen, dass die Nachbarn von X ihre 2-hop Nachbarschaft nur teilweise kennen und die Menge der MPRs ebenfalls unzureichend ist. Fraglich ist die Tragweite dieses Angriffes. Die Funktionalität des Routings wird höchstens in dem Maße eingeschränkt, wie es auch durch den Wegfall des angreifenden Knotens X der Fall wäre.

3.3.4 Identity und Link Spoofing mit Generierung falscher TC-Nachrichten

Ein Angreifer X kann durch TC-Nachrichten Information über die Nachbarschaft von Knoten Z, der Ziel des Angriffs ist, im Netz verbreiten. Beim Identity Spoofing wird X dies im Namen

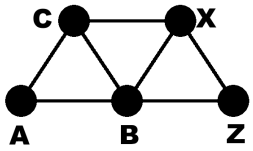

Abbildung 4: X propagiert im Namen von A dessen Nachbarschaft zu Z

eines geeignet gewählten Knotens tun, der tatsächlich keine Nachbarschaft zu Z hat. Im Beispiel (vgl. Abbildung 4) wird von X eine Nachbarschaft von A und Z angegeben. Andere Knoten im Netz berechnen daraufhin falsche Routen zu Z, im Beispiel würde C seine Pakete an Z über A senden, welcher diese nicht weiterleitet.

Ebenso kann X durch standardkonformes Verhalten zum MPR gewählt werden, dann aber keine oder unvollständige TC-Nachrichten versenden. Dadurch können die Knoten, die ihn als MPR gewählt haben, isoliert werden.

3.3.5 MID/HNA Angriff

MID und HNA Nachrichten beinhalten Informationen über Knoten mit mehreren Interfaces. Angreifer X kann durch entsprechende Nachrichten die Information verbreiten, ein bestimmter Knoten (Identity Spoofing) oder er selbst (Link Spoofing) hätten bestimmte weitere Interfaces. Dadurch werden Informationen zu diesen Interfaces falsch geroutet.

3.3.6 ANSN Angriff

In OLSR beinhaltet jede TC-Nachricht die vom Absender pro Nachricht inkrementierte Sequenznummer ANSN. Mit Hilfe dieser Nummer werden alte von aktuellen Nachrichten unterschieden - eine TC-Nachricht mit einer ANSN macht jede TC-Information mit älterer ANSN ungültig. Angreifer X kann im Namen eines anderen Knotens A eine TC-Nachricht im Netz verbreiten, die eine viel zu hohe ANSN-Nummer enthält. Alle TC-Informationen, die A danach mit zu niedrigem ANSN absendet, werden von anderen Knoten verworfen.

3.3.7 Blackhole Angriff

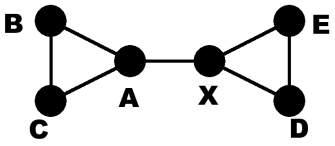

Abbildung 5: Blackhole Angriff: X leitet keine TC-Nachrichten von A weiter an D und E

Wird eine TC-, MID- oder HNA- Nachricht in Netzen ohne Redundanz durch einen Angreifer X nicht weitergeleitet, so berechnen einige Knoten die Routen auf der Basis lückenhaften Wissens. Es kann dazu kommen, dass ein Knoten keine Route zu einem anderen Knoten berechnen kann, da auf jeder möglichen Route vom Start- zum Zielknoten mindestens ein Link nicht bekannt ist. Im Beispiel in Abbildung 5 leitet X keine TC-Nachrichten an D und E weiter. Dadurch kennen diese die Verbindung zwischen A, B und C nicht und können weder mit B noch mit C kommunizieren.

3.3.8 Replay Angriff

Ein Angreifer kann durch die Wiederholung früherer Nachrichten veraltete Topologieinformationen im Netz propagieren. Damit die alten Nachrichten nicht verworfen werden, muss der Angreifer die Nachrichten mit neuen Sequenznummern versehen. Bei HELLO-Nachrichten ist dies die MSN, bei TC-Nachrichten zusätzlich die ANSN. Im Falle eines Replay-Angriffes mit Hilfe von TC-Nachrichten findet automatisch auch ein ANSN Angriff statt. Ein Replay Angriff im klassischen Sinne auf OLSR ist nicht möglich.

3.3.9 Wormhole Angriff

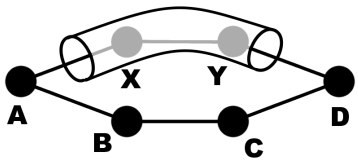

Abbildung 6: Wormhole Angriff: X und Y bilden einen für A und D nicht sichtbaren Tunnel

Ein Wormhole ist ein Pfad zwischen zwei Knoten, der durch einen oder mehrere Angreifer zwischen diesen Knoten genutzt wird um Kontrollnachrichten transparent weiterzuleiten. Im Beispiel aus Abbildung 6 bilden X und Y ein Wormhole zwischen den Knoten A und D. Der Angreifer X ist hierbei ein Nachbar von A, der die Kontrollnachrichten von A durch einen für A und D transparenten Tunnel zu D weiterleitet. Im Beispielfall besteht dieser Tunnel aus den Knoten X und Y. Bei diesem Angriff werden durch dieses Wormhole HELLO-Nachrichten weitergeleitet. Der vermeintlich direkte Link zwischen A und D kann daraufhin vom Angreifer beliebig kontrolliert werden.

Der nicht existente Link wird bei der Berechnung der Routingtabellen berücksichtigt. Besonders kritisch sind lange Wormholes, deren Tunnel aus vielen Knoten bestehen und die somit entfernte Knoten verbinden. Die vermeintlichen Links werden vom Routing häufig berücksichtigt, der Verkehr wird also bevorzugt über den Angreifer geleitet. Dieser kann dann umso mehr Verbindungen kontrollieren.

3.3.10 MPR Angriff

Im Normalfall prüft ein Knoten vor der Weiterleitung von Topologieinformationen, ob der sendende Knoten ihn als MPR gewählt hat, nur dann werden entsprechende Nachrichten weitergeleitet. Beim MPR Angriff leitet ein nicht-MPR Nachrichten weiter, verhält sich also wie ein MPR. In bestimmten Konstellationen wird ein benachbarter MPR diese Nachrichten nicht weiterleiten, da er davon ausgeht, dass die Information bereits allen Knoten, die ihn als MPR gewählt haben, bekannt ist.

3.4 Entdeckung von Angriffen

Eine Auswertung von OLSR-Paketen und den enthaltenen Nachrichten kann helfen, Angriffe auf das Routing zu entdecken. Nicht jeder Angriff kann mit Sicherheit erkannt werden. Ob tatsächlich ein Angriff vorliegt und wer Angreifer oder Ziel ist, kann nicht immer zweifelsfrei festgestellt werden. Wurde ein Angriff erkannt, so bedeutet dies nicht, dass diesem Angriff begegnet werden kann.

Knoten können analysieren, welche Information über sie selbst oder ihre Nachbarschaft im Netz verbreitet werden. Kommt es dabei zu Abweichungen von den eigenen Informationen, so ist dies ein Indiz für einen Angriff:

- Empfängt ein Knoten eine HELLO-Nachricht mit der eigenen Adresse als Absenderkennung, so kann diese Nachricht ein Hinweis auf einen der folgenden Angriffe sein:
  1. Identity Spoofing Angriff auf Basis falscher HELLO-Nachrichten (vgl. Abschnitt 3.3.1). Die HELLO-Nachricht wird nur empfangen, wenn der Angreifer ein direkter Nachbar des Knotens ist. In der wahrscheinlicheren Konstellation, also wenn der Angreifer ein entfernter Knoten ist, wird dieser Angriff so nicht erkannt.
  2. Wird in der Nachricht das Feld WILLINGNESS mit WILL NEVER angegeben, so deutet dies auf einen Identity Spoofing Angriff hin, der auf eben dieser falschen WILLINGNESS-Information basiert (vgl. Abschnitt 3.3.2). Ziel des Angriffes ist einer der Nachbarknoten.
  3. Auch ein Replay-Angriff auf Basis von HELLO-Nachrichten kann so erkannt werden (vgl. Abschnitt 3.3.8).
  4. Wormhole-Angriffe können entdeckt werden, sofern der Knoten eigene HELLO-Nachrichten empfängt, wenn diese am Ausgang des Wurmloches wieder versendet werden. Je kürzer das Wormhole (besteht es zum Beispiel nur aus einem angreifenden Knoten), desto wahrscheinlicher ist dies. Ist das Wormhole ausreichend groß und werden HELLO-Nachrichten zwischen den Endknoten des transparenten Tunnels entsprechend umkodiert, so ist eine Erkennung auf diese Art nicht möglich (vgl. Abschnitt 3.3.9).
- Eine TC-Nachricht kann auf drei verschiedene Angriffe hindeuten:
  1. Wird in dieser Nachricht X als MPR eines Knotens A angegeben, wobei A jedoch tatsächlich keine Verbindung zu X hat, so kann dies ein Hinweis auf einen Identity Spoofing Angriff auf Basis falscher HELLO-Nachrichten sein (vgl. Abschnitt 3.3.1). Der angegebene MPR X ist in diesem Fall der Angreifer. Der kombinierte Identity und Link Spoofing Angriff (vgl. Abschnitt 3.3.4) führt zu ähnlichen Nachrichten.
  2. Enthält eine TC-Nachricht eine ungewöhnlich hohe ANSN, so nutzt der Angreifer diese eventuell im Rahmen eines ANSN Angriffes aus (vgl. Abschnitt 3.3.6). Ein ähnlicher Effekt entsteht jedoch auch bei länger andauernden Störungen der physikalischen Verbindung.
  3. Empfängt ein Knoten eine eigene, alte Nachricht, so ist dies ein Hinweis auf einen Replay-Angriff (vgl. Abschnitt 3.3.8).
- Gibt ein Nachbar eines Knotens eine existierende Verbindung nicht in seinen HELLO-Nachrichten an, so ist dies ein Hinweis auf eine Variante des Link Spoofings durch Angabe falscher Nachbarschaften (vgl. Abschnitt 3.3.3). Um dies zu erkennen, muss eine entsprechendes HELLO-Nachricht vom Kommunikationspartner des Angreifers empfangen werden, was nicht immer möglich ist.
- Entdeckt ein Knoten A, dass einer seiner MPRs nicht die gemäß Standard erforderlichen TC-Nachrichten versendet und damit A isoliert, so führt dieser MPR eventuell eine Variante des kombinierten Identity und Link Spoofing Angriffes aus (vgl. Abschnitt 3.3.4).
- Empfängt ein Knoten inkorrekte MID oder HNA Nachrichten, in denen er selbst als Absender beschrieben wird, so deutet dies auf einen MID- bzw. HNA Identity Spoofing Angriff hin (vgl. Abschnitt 3.3.5).

- Stellt ein Knoten fest, dass ein Knoten, den er nicht zum MPR gewählt hat, die gesendeten TC-Nachrichten weiterleitet, so kann es sein, dass dieser Knoten einen MPR Angriff durchführt (vgl. Abschnitt 3.3.10).

Weiterhin können Knoten die Plausibilität der empfangen Informationen prüfen. Beispiele für derartige Untersuchungen könnten übersprungene ANSNs von Knoten (Indiz für einen ANSN Angriff, siehe auch Abschnitt 3.3.6) oder ungewöhnlich lange Latenzen sein (Indiz für einen Wormhole Angriff, siehe auch Abschnitt 3.3.9 und ein entsprechendes Verfahren in [HoHF05]).

Im Umfeld von MANETs ist davon auszugehen, dass die einzelnen Knoten begrenzte Rechen- und Speicherkapazität haben. Es ist zu vermuten, dass umfangreiche statistische Analysen sowie große Modelle der geographischen Verteilung der Knoten Angriffe gut erkennen können. Es ist jedoch zu untersuchen, ob die Kapazität der mobilen Knoten für derartige Methoden ausreicht. Entsprechende Tools werden in [Raff05] aufgeführt.

4 Secure OLSR

Secure OLSR wird in [ACJL+03, AdRM04] beschrieben. Eine ausführliche Darstellung findet sich ebenfalls in [Raff05].

Der wesentliche Ansatz von Secure OLSR ist, HELLO- und TC- Nachrichten zu signieren und damit deren Integrität zu sichern. Das entsprechende Verfahren wird in Abschnitt 4.1 beschrieben. Weiterhin werden die Kontrollnachrichten mit einem Zeitstempel versehen, Abschnitt 4.2 erklärt das hierzu benötigte Protokoll zur Synchronisation der Uhren. Die Signatur wird durch ein kryptographisches System abgesichert. Hierfür können sowohl gemeinsame Schlüssel (symmetrische Verschlüsselung) oder asymetrische Verfahren eingesetzt werden. In Abschnitt 4.3 wird eine Public Key Infrastruktur (PKI) vorgestellt, die eine entsprechende Verschlüsselung leistet. Durch die Verwendung einer PKI können sowohl die Authentifizierung der Absender als auch die Nicht-Abstreitbarkeit der Nachrichten sichergestellt werden. Eine Bewertung, inwiefern die präsentierten Maßnahmen Angriffe auf OLSR abwehren, wird in Abschnitt 4.4 vorgenommen.

4.1 Signaturnachrichten

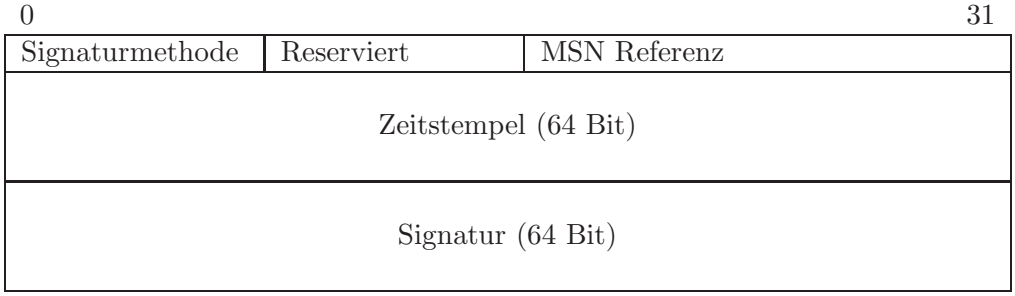

Tabelle 1: Paketformat der Signaturnachricht in Secure OLSR

Wesentliche Erweiterung in Secure OLSR ist ein neuer Nachrichtentyp, der Signaturen aufnimmt. Eine Signatur berechnet sich aus der zu signierenden Nachricht (HELLO oder TC) bzw. einem Haswert einer solchen Kontrollnachricht, einem Zeitstempel sowie einem gemeinsamen geheimen Schlüssel. Verschiedene Signaturverfahren können eingesetzt werden. Als

Wert in den Feldern Hop Count und Time-To-Live (TTL) wird jeweils 0 angenommen, da sich diese Werte bei der Weiterleitung der Nachricht ändern und dadurch eine entsprechende Signatur ungültig würde.

Das Format der Signatur ist in Tabelle 1 angegeben. Die Elemente der Signaturnachricht sind:

Signiermethode (8 bit): Hier wird codiert, nach welcher Methode (aus einer Menge vorgegebener Methoden) signiert wird, also welcher Schlüssel, welches Hashverfahren und welches Zeitstempelformat verwendet wird.

Reserviert (8 bit)

MSN Referenz (16 bit): Verweis auf die zu signierende Nachricht.

Zeitstempel (64 bit): Zeit der Signierung

Signatur (64 bit): Signatur der Kontrollnachricht mit der angegebenen MSN sowie des Zeitstempel, unter Verwendung der Signiermethode, die im entsprechenden Feld beschrieben wird.

Um eine Nachricht zu signieren, führt der Autor einer Kontrollnachricht folgende Schritte durch:

1. Generierung der Nachricht, setzen des Hop Counts und der TTL auf 0, Bestimmung der MSN der Nachricht
2. Erfassung der Zeit und Bildung des aktuellen Zeitstempels
3. Berechnung der Signatur der Nachricht oder des Hashwertes der Nachricht sowie des Zeitstempels mit dem gemeinsamen, geheimen Schlüssel
4. Zusammensetzung der Signaturnachricht und Versand

Der Empfänger einer Kontrollnachricht speichert eingegangene Kontrollnachrichten, und führt bei Bedarf folgende Schritte aus:

1. Vergleich von Absender und MSN der signierten Nachricht mit den Angaben in der Signaturnachricht
2. Setzen von Hop Count und TTL der Nachricht auf 0
3. Verifikation der Signatur unter Berücksichtigung der Angaben zur Signiermethode und des Zeitstempels
4. Prüfung der Aktualität des Zeitstempels der Nachricht

4.2 Zeitstempel

Die Verwendung von signierten Zeitstempeln schützt das Protokoll vor Replay Angriffen. Problematisch ist jedoch die Überprüfung eines fremden Zeitstempels, da dies synchrone Uhren voraussetzt. Verschiedene Verfahren zum Austausch von Zeitstempeln werden in [ACJL+03] beschrieben. Das in Secure OLSR verwendete Protokoll basiert auf dem Public Key Verfahren von Needham und Schroeder [NeSc78].

Bei dem verwendeten Verfahren schickt zunächst ein Knoten A seinen aktuellen (t_0) Zeitstempel T_A sowie seine eigene Adresse an B. Diese Information signiert A mit seinem privaten Schlüssel (Sa). B antwortet mit einer Nachricht, die sowohl den Zeitstempel von A und dessen Adresse als auch einen neueren Zeitstempel ($T_B(t_1)$) sowie die eigene Adresse beinhalten, und die mit Sb signiert wurde. Wiederum antwortet A mit einer entsprechenden Nachricht. Diese letzte Nachricht verhindert Replay Angriffe, indem sie sich auf eine vorher von B gesendete Nachricht bezieht.

$1.Nachricht : A \xrightarrow{(A,T_A(t_0))_{Sa}} B$

$2.Nachricht : A \xleftarrow{(A,T_A(t_0),B,T_B(t_1))_{Sb}} B$

$3.Nachricht : A \xrightarrow{(A,T_A(t_2),B,T_B(t_1))_{Sa}} B$

In [ACJL+03] wird darüberhinaus eine Verbesserung dieses Verfahrens beschrieben. Ein neuer Nachrichtentyp zum Zeitstempelaustausch wird definiert. Die Knoten fluten ihre Zeitstempel mit Hilfe dieser Nachrichten ins Netz, wodurch der Kommunikationsaufwand im Netz reduziert wird, da der Umfang der paarweisen Zeitstempelaustausche abnimmt.

4.3 Public Key Infrastruktur

Secure OLSR setzt Mechanismen zum Austausch von Schlüsseln voraus. Dies kann ein symmetrisches Verfahren oder eine Public Key Infrastruktur (PKI) leisten. In [ACJL+03] werden zwei PKIs für MANETs vorgeschlagen, eine proaktive und eine reaktive Variante. Das reaktive Verfahren wird an dieser Stelle nicht näher beschrieben, da es beim proaktiven Design von OLSR nicht angebracht scheint. Bei einem proaktiven Verfahren wird der Aufwand minimiert, der in dem Moment betrieben werden muss, wenn eine Kontrollnachricht versendet werden soll. Der dadurch schnellere Versand von Kontrollnachrichten rechtfertigt den insgesamt gestiegenen Overhead durch die ständig anfallenden Schlüsselaustausche.

Die proaktive PKI führt drei Klassen ein, in die jeder Knoten die anderen Knoten einteilt.

Signierstellen: Ihre öffentlichen Schlüsseln sind den Knoten bereits vor der Initialisierung bekannt. Sie signieren öffentliche Schlüssel von zugelassenen Knoten. Sie fluten signierte Listen von öffentlichen Schlüsseln von zugelassenen Knoten ins Netz.

Vertrauenswürdigen Knoten: Dazu zählen alle Knoten, deren öffentliche Schlüssel von einer Signaturstelle signiert wurden.

Nicht vertrauenswürdige Knoten: Dies sind Knoten, deren öffentlicher Schlüssel entweder nicht bekannt ist oder bei denen dieser nicht durch eine Signierstelle beglaubigt wurde.

Um eine Initialisierung der Netze zu ermöglichen, ist eine Anpassung notwendig. Bestimmte Nachrichten müssen auch unsigniert angenommen werden, damit die Signierstellen die Liste der zugelassenen Knoten in das Netz fluten kann. Hierzu werden jedoch nicht vertrauenswürdige und vertrauenswürdige Knoten getrennt behandelt, so können nur vertrauenswürdige Nachbarn zu MPRs gewählt werden. Indem HELLO-Nachrichten von nicht vertrauenswürdigen Nachbarn zugelassen werden, wird bei der Initialisierung von den Signierstellen ausgehend das gesamte Netz in die PKI einbezogen.

Der Netzaufbau wird durch dieses Verfahren verzögert. Außerdem ist die Wahl der MPRs zwischenzeitlich suboptimal, da nur die Menge der vertrauenswürdige Knoten als MPR in Frage kommt - diese Menge wächst im Laufe der Initialisierung. Letztendlich werden jedoch alle legitimen Knoten des Netzes von den Signaturstellen bekanntgegeben. Voraussetzung für das Funktionieren der PKI ist, dass die Signaturstellen eine Liste der legitimen Knoten hat und das jeder Knoten bei der Initialisierung bereits die Unterschrift mindestens einer Signaturstelle überprüfen kann.

4.4 Bewertung

Durch die Kombination der beschriebenen Verfahren wird die Sicherheit des Routings von OLSR erhöht.

Auf das Routing bezogen wird keine Geheimhaltung gewährleistet, da Kontrollnachrichten nach wie vor unverschlüsselt übertragen werden. Die Signatur der Kontrollnachrichten garantieren jedoch deren Integrität, Nicht-Abstreitbarkeit sowie eine Authentifizierung des Absenders. Das Routing von Secure OLSR berücksichtig nur Knoten, die von der Signaturstelle authorisiert wurden, indem ihre öffentlichen Schlüssel signiert und publiziert wurden.

Die in Abschnitt 3.3 beschriebenen Angriffe richten sich gegen die Verfügbarkeit des Netzes. Dieser Aspekt von Sicherheit wird durch Secure OLSR ebenfalls berücksichtigt. Offensichtlich sind alle Varianten des Identity Spoofing unmöglich. Da die Signatur auch die MSN der Nachrichten umfasst, werden Replay Angriffe ebenfalls verhindert. Diese beiden Klassen von Angriffen werden sogar verhindert, wenn der Knoten legitimer Teilnehmer des Netzes ist und einen von einer Signaturstelle signierten öffentlichen Schlüssel besitzt.

Besitzt ein angreifender Knoten einen signierten öffentlichen Schlüssel, so kann dieser Knoten weiterhin das Routing von OLSR stören. Dies ist durchaus realistisch, da zum Beispiel durch Viren legitime Knoten für Angriffe eingesetzt werden können. Wird ein Knoten jedoch nicht von einer Signaturstelle authorisiert, so werden die von ihm generierten oder manipulierten Kontrollnachrichten ignoriert.

Nur einer der beschriebenen Angriffe ist durch einen solchen Knoten noch möglich. Der Wormhole Angriff verlangt weder die Manipulation noch die Generierung von Nachrichten. Es bleibt zu untersuchen, ob Secure OLSR dahingehend erweitert werden kann, das Wormhole Angriffe unmöglich werden. Die Weiterleitung von Kontrollnachrichten führt jedoch unter Umständen zu Übertragungsverzögerungen, insbesondere bei langen Wormholes. Dies kann auf zweierlei Weise zumindest für die Erkennung von Wormholes genutzt werden:

1. Exakt synchronisierte Uhren können die Paketlaufzeit feststellen. Dazu wird beim Empfänger der signierte Zeitstempel mit der eigenen Zeit verglichen. Allerdings setzt dies voraus, dass der Empfänger die Nachricht sehr schnell oder nur in einer exakt vorgegebenen Zeit signieren kann. Außerdem ist die Genauigkeit der Uhren der mobilen Knoten begrenzt. Daher ist anzunehmen, dass dieses Verfahren nur große Laufzeitunterschiede feststellen kann.

2. In [HoHF05] wird die Laufzeit von Paketen mittels eines Ping-Pong Tests festgestellt. Die Autoren beschreiben, wie Wormholes dadurch erkannt werden können. Bei diesem Verfahren werden die Ping- und Pong Nachrichten aus Verzögerungsgründen nicht signiert, was die Wirksamkeit des Verfahrens fraglich erscheinen lässt: Der oder die Angreifer sind Nachbarn der angegriffenen Knoten und können somit auf eine Ping-Nachricht reagieren, ohne den privaten Schlüssel des vermeintlichen Kommunikationspartners zu kennen.

5 Related Work

Dieser Arbeit lagen vor allem die Arbeiten des Projektes Hipercom vor. Hierzu zählen nicht nur der entsprechende RFC ([ClJa03]), sondern auch die Dissertation von Daniele Raffo ([Raff05]). Secure OLSR in der hier vorgestellten Version wurde ebenfalls dort entwickelt ([ACJL+03], [AdRM04]). Ebenso wurde dort eine Lösung auf Basis von Ortsinformationen der Knoten entwickelt (vgl. [RACM05]).

Mit Secure OLSR vergleichbare Lösungen wurden weiterhin sowohl in [HTRA+04] als auch in [HoHF05] vorgestellt.

Einen Überblick über zahlreiche weitere Ansätze zum sicheren Routing in MANETs bietet [HuPe04].

6 Fazit und Ausblick

OLSR kann vor allem in großen und dichten MANETs effizient zur Routenberechnung eingesetzt werden. Allerdings bietet OLSR in seiner momentanen Fassung keinerlei Sicherheit gegen eine Vielzahl von Angriffen. Verschiedene solcher Angriffe auf Routing in MANETs und insbesondere auf OLSR wurden vorgestellt.

Weiterhin wurden einige Indizien präsentiert, mit denen Knoten Angriffe entdecken können. Es bleibt zu untersuchen, ob insbesondere mobile und damit leistungsschwache und batteriebetriebene Geräte zu den entsprechenden statistischen Analysen fähig sind, um Angriffe zu erkennen.

Secure OLSR ist eine Antwort auf die Sicherheitsfragen, die im Zusammenhang mit OLSR aufgeworfen werden. Wesentliche Prinzipien von Secure OLSR wurden präsentiert. Mit wenigen Eingriffen bzw. einer Erweiterung des OLSR Standards wurde eine Möglichkeit geschaffen, OLSR-Nachrichten zu signieren. Die in Secure OLSR für den Austausch von Zeitstempeln und geheimen Schlüsseln eingesetzten Verfahren wurden vorgestellt.

Secure OLSR bietet zwar keine Geheimhaltung der Routinginformationen, die in den Kontrollnachrichten versendet werden, es sichert aber die Integrität und Nicht-Abstreitbarkeit von Kontrollnachrichten und authentifiziert den Absender. Die Verfügbarkeit des Routings kann dadurch im Hinblick auf die meisten der vorgestellten Angriffe erhöht werden.

Einen Großteil der beschriebenen Angriffe, nämlich alle auf Identity Spoofing basierenden, kann Secure OLSR abwehren. Es ist dennoch denkbar, dass beispielsweise durch Viren legitime Knoten für Angriffe verwendet werden können. Geht man allerdings davon aus, dass einmal legitimierte Knoten keine Angriffe ausführen, so werden alle Angriffe bis auf Wormhole Angriffe wirksam verhindert. Es wurden zwei Ansätze aufgezeigt, wie Wormholes entdeckt werden können. Es bleibt zu untersuchen, ob eines dieser Verfahren wirksam zum Schutz vor Wormhole Angriffen eingesetzt werden kann.

Literatur

[ACJL+03] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Mühlenthaler und D. Raffo. Securing the OLSR protocol. In *Proceedings of the Second Annual Mediterranean Ad Hoc Networking Workshop 2003*, Mahdia, Tunisia, Juni 2003. IFIP.

[AdRM04] C Adjih, D. Raffo und P. Mühlethaler. Attacks Against OLSR: Distributed Key Management for Security. In *Proceedings of the OLSR Interop and Workshop*, San Diego, CA, USA, August 2004. IETF.

[Clau05] T. Clausen. The Optimized Link-State Routing Protocol version 2. Internet Draft draft-ietf-manet-olsrv2-00, August 2005.

[ClJa03] T. Clausen und P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), Oktober 2003.

[HoHF05] F. Hong, L. Hong und C. Fu. Secure OLSR. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, Band 1, Taipei, Taiwan, März 2005. IEEE Computer Society, S. 713–718.

[HTRA+04] A. Hafslund, A. Tonnesen, R. Bjorgum Rotvik, J. Andersson und O. Kure. Secure Extension to the OLSR protocol. In *Proceedings of the OLSR Interop and Workshop*, San Diego, CA, USA, August 2004. IETF.

[HuPe04] Y. Hu und A. Perrig. A Survey of Secure Wireless Ad Hoc Routing. *IEEE Security and Privacy: Special issue on Making Wireless Work* 2(3), Mai/Juni 2004, S. 28–39.

[NeSc78] R. Needham und M. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM* 21(12), 1978, S. 993–999.

[RACM05] D. Raffo, C. Adjih, T. Clausen und P. Mühlethaler. Securing OLSR Using Node Locations. In *Proceedings of the European Wireless Conference 2005*, Nicosia, Zypern, April 2005.

[Raff05] D. Raffo. *Security Schemes for the OLSR Protocol for Ad Hoc Networks*. Dissertation, Université Paris 6, Pierre Et Marie Curie, September 2005.

Abbildungsverzeichnis

Tabellenverzeichnis

Secure Ad-hoc on Demand Distance Vector Routing

Thomas Heilbronner

Kurzfassung

Mit der zunehmenden Verbreitung mobiler Endgeräte werden Sicherheitsaspekte in MANETs zukünftig eine bedeutende Rolle in der Forschung einnehmen. Es wird am Beispiel AODV [PeBRD03] gezeigt, wie ein mehrfach implementiertes Protokoll nachträglich um Sicherheitsaspekte erweitert wird. Der bei der IETF als Draft vorliegende Entwurf Secure AODV (SAODV [Zapa05]) spielt hierbei eine zentrale Rolle. Die zur Absicherung der AODV Routing-Nachrichten eingesetzten Mechanismen werden vorgestellt und in ihrer Funktionsweise erläutert.

1 Einleitung

1.1 Motivation

Im Vergleich zu Mobilen Ad hoc Netzen (2.1) haben wir im Corebereich eine statische Topologie. Hier „kennt“ man seine Nachbarn genau und kann sich durch relativ einfache Mechanismen gegenüber Angriffen auf das Routing absichern. Aber mit zunehmender Mobilität müssen neue Netze geschaffen werden, die sich einer dynamischen, sich ständig im Fluss befindenden Topologie anpassen. Es ist jedem jederzeit möglich an einer Kommunikation teilzunehmen, Dienste von Dritten in Anspruch zu nehmen oder selbst anzubieten. Diese sehr offene Architektur ist in vielen Fällen genau das, was erwünscht ist. Die Vorteile beim Aufbau derartiger Netze ermöglichen Problemlösungen, die auf anderem Wege nicht zu realisieren oder schlichtweg zu teuer sind. Deshalb kommt es hierbei auch häufig zum Einsatz von drahtlosen Technologien, die es potenziellen Angreifern sehr einfach machen, auf dem Medium mitzuhören bzw. zu senden. Mit dieser Offenheit ergeben sich aber leider auch neue Probleme, zum Beispiel in Zuverlässigkeit, Dienstgüte und Sicherheit.

Wie bei so vielen Technologien liegt der Ursprung in einer militärischen Nutzung, so auch bei Ad hoc Netzen. Eine eigene stationäre Infrastruktur aufzubauen und zu unterhalten ist kostspielig und beansprucht Zeit. Daher werden Ad-Hoc Netze als Mittel gesehen, die Kommunikation zwischen einzelnen Verbänden zu ermöglichen. In einem derartigen Szenario wäre es fatal, wenn es zu einer Störung oder Verfälschung der übertragenen Informationen kommen würde. Deshalb müssen alle Kanäle, von der physikalischen Schicht über Routing-Informationen bis hin zu den zu übertragenden Nutzdateneinheiten gegenüber Angriffen geschützt werden.

Aber auch im kommerziellen Bereich werden Ad-Hoc Netze immer interessanter. Die zunehmende Verbreitung mobiler Endgeräte ermöglicht eine Reihe neuer Dienstleistungen, die den Netzbetreibern neue Einnahmequellen versprechen. Um aber eine Dienstleistung zuverlässig anbieten zu können, muss ein gewisser Schutz vor externen Angriffen gegeben sein, der diese Angriffe wenn nicht verhindert, dann doch stark erschwert bzw. in ihren Auswirkungen lokal begrenzt.

1.2 Gliederung

In Kapitel 2 werden die für SAODV grundlegenden Techniken vorgestellt und kurz erläutert. Eine Einführung in die Funktionsweise des Ad Hoc on Demand Distance Vector (AODV) Routing Protokolls wird in Kapitel 3 gegeben. In Kapitel 4 wird dann ausführlich auf Secure AODV eingegangen. Die Zusammenfassung in Kapitel 5 schließt die Betrachtung ab.

2 Grundlagen

In diesem Kapitel werden die wesentlichen Techniken vorgestellt, die zum Verständnis von SAODV benötigt werden. Zunächst wird der Begriff des Ad Hoc Netzes (s. 2.1) erklärt. In den folgenden Abschnitten wird eine kurze Einführung zum Secure Hash Algorithmus (s. 2.2), in die Funktionsweise von Hash Chains (s. 2.3) und das RSA Kryptosystem (s. 2.4) gegeben.

2.1 Mobile Ad-Hoc Netze (MANETs)

Ein Ad Hoc Netz zeichnet sich durch die nicht benötigte Infrastruktur aus. Kommunikation zwischen beteiligten Knoten findet entweder direkt oder über Dritte statt. Direkt immer dann, wenn sich Sender und Empfänger innerhalb ihrer maximalen Sendereichweite aufhalten. Befinden sie sich nicht innerhalb ihrer maximalen Sendereichweiten erfolgt der Datentransport über Zwischenknoten, die sich ebenfalls im Ad Hoc Netz befinden. Jeder Knoten muss dazu in der Lage sein als Router zu fungieren. Das große Problem dabei ist, dass sich die Knoten innerhalb von Ad Hoc Netzen meist bewegen, die Topologie des Netzes ändert sich also ständig. Dieses Szenario stellt vollkommen neue Anforderungen an die verwendeten Routingprotokolle. Durch die ständigen und spontanen Änderungen, das Fehlen von Infrastruktur und zentraler Komponenten innerhalb des Netzes, treten Sicherheitsprobleme auf, die man so aus strukturbasierten Netzen nicht kennt. Die AODV Erweiterung SAODV versucht diesen veränderten Umständen Rechnung zu tragen.

2.2 Sichere Hash Algorithmen

Mit Secure Hash Algorithm [oSTe02] wird eine ganze Gruppe standardisierter, kryptografischer Hash Funktionen bezeichnet. Eine Hash Funktion liefert aus einer Eingabe der Länge n eine Ausgabe der Länge m, und im Regelfall gilt $n > m$. In diesem Zusammenhang wird von sicheren Hash Funktionen gesprochen, wenn es für den jeweils gegebenen Algorithmus rechnerisch nahezu unmöglich ist,

1. aus dem Hash-Wert die ursprüngliche Nachricht zu rekonstruieren

2. aus zwei unterschiedlichen Nachrichten denselben Hash-Wert zu erzeugen, d.h. die Funktion ist kollisionsfrei. Jede noch so geringe Veränderung der Nachricht wird mit einer sehr hohen Wahrscheinlichkeit zu einem unterschiedlichen Hash-Wert führen

Beispiel — SHA-1 [EaJo01] berechnet aus 512 bit Klartextblöcken einen 160 bit Hash-Wert und das für Nachrichten bis zu einer Gesamtlänge von 2^{64} bit.

2.3 Hashketten

Hashketten bauen auf einer so genannten „Einwegfunktion“ h auf. Eine Hashkette der Länge N berechnet sich nun durch N-maliges Anwenden der Hash-Funktion h auf einen Ausgangswert s (seed).

$$\mathsf{h}^{\mathsf{N}}(s) = \mathsf{h}(\mathsf{h}(\ldots(\mathsf{h}(s)))) \quad (\mathsf{N} - mal)$$

Hashketten lassen sich zur effizienten Authentisierung einsetzen. Aus $\mathsf{h}^{\mathsf{N}}(s)$ lässt sich $\mathsf{h}^{\mathsf{N}-1}(s)$ nicht berechnen ohne s zu kennen, wobei sich die Korrektheit von $\mathsf{h}^{\mathsf{N}-1}(s)$, durch erneutes Anwenden von h und einen anschließenden Vergleich ermitteln lässt. Damit dieser Vergleich stattfinden kann, muss $\mathsf{h}^{\mathsf{N}}(s)$ auf sicherem Weg an alle Kommunikationspartner verteilt werden.

2.4 RSA

Bei den Ausführungen zu RSA handelt es sich um eine sinngemäße Übersetzung von [Labo]. RSA [RiSA78][JoKa03] wurde 1977 von Ron Rivest, Adi Shamir und Leonard Adleman entwickelt. Bei RSA handelt es sich um ein asymmetrisches Kryptosystem, es gibt also einen öffentlichen und einen privaten Schlüssel. Zur Erzeugung werden zunächst zwei große Primzahlen p und q gewählt. Durch die Multiplikation von p und q erhält man n. Man wählt eine Zahl e die kleiner als n und teilerfremd zu $(p-1)$ und $(q-1)$ ist. Nun sucht man eine Zahl d, so dass $(e*d)-1$ durch $(p-1)*(q-1)$ teilbar ist. Die Werte e und d werden öffentlicher bzw. privater Exponent genannt. Der öffentliche Schlüssel wird nun aus dem Paar (n,e) und der private Schlüssel aus dem Paar (n,d) gebildet. Die Sicherheit von RSA ist an die Annahme geknüpft, das das Faktorisieren großer Zahlen schwierig ist. Sollte es möglich werden große Zahlen in relativ kurzer Zeit zu Faktorisieren, würde dies RSA „brechen“.

Im Folgenden wird beschrieben, wie RSA zur Geheimhaltung und Authentifizierung eingesetzt werden kann.

RSA Verschlüsselung

Alice möchte eine Nachricht m an Bob schicken. Alice erstellt den Chiffretext c durch Potenzieren: $c = m^e \bmod n$, wobei (n,e) Bobs öffentlicher Schlüssel ist. Sie sendet c an Bob. Um c zu dechiffrieren potenziert Bob ebenfalls: $m = c^d \bmod n$. Die mathematische Beziehung zwischen e und d garantiert, das Bob m korrekt wiederherstellen kann. Da nur Bob d kennt, kann nur Bob c entschlüsseln.

RSA Authentifizierung

Alice möchte eine Nachricht m an Bob unterschreiben, so das Bob sicher sein kann, dass diese Nachricht authentisch und von Alice ist. Alice erstellt eine elektronische Unterschrift durch Potenzieren: $s = m^d \bmod n$, wobei (n,d) der private Schlüssel von Alice ist. Sie sendet m und s an Bob. Dieser prüft die Unterschrift durch Potenzieren: $s^e \bmod n$ muß mit m übereinstimmen, wenn (n,e) Alice öffentlicher Schlüssel ist. Um den Aufwand bei der Ent- und Verschlüsselung zu reduzieren, wird normalerweise nicht die gesamte Nachricht m signiert, sondern nur ein Hash-Wert $\mathsf{h}(m)$ von m. Der signierte Wert von $\mathsf{h}(m)$ wird dann zusammen mit m übertragen.

Damit ist die Verschlüsselung und Authentifizierung ohne ein gemeinsames Geheimnis möglich: Jede Person hat nur die öffentlichen Schlüssel aller Beteiligten und den eigenen privaten Schlüssel. Jeder kann Nachrichten verschlüsselt senden oder Unterschriften prüfen, aber nur derjenige, der den richtigen privaten Schlüssel hat, kann Nachrichten entschlüsseln oder unterschreiben.

3 Ad-Hoc on Demand Distance Vector (AODV) Routing

Bei Secure AODV [Zapa05] handelt es sich um eine Erweiterung von AODV [PeBRD03] und darum wird an dieser Stelle kurz auf die Funktionsweise von AODV eingegangen. Abschnitt 3.1 gibt allgemeine Informationen zu AODV. In Abschnitt 3.2 wir auf die einzelnen Phasen Routensuche, Routenpflege und Routenlöschung eingegangen. Die Einträge der AODV Routing Tabelle finden sich in Abschnitt 3.3.

3.1 AODV Überblick

Wie der Name schon nahelegt, handelt es sich bei AODV um ein Distanz Vector Routing-Protokoll. Das bedeutet, in der Routing Tabelle (Tabelle 3.3) steht neben weiteren Informationen für jeden bekannten Ziel-Knoten ein Vektor aus Entfernung (hopcount) und nächstem Knoten (nexthop). Bei AODV kommt für jeden Knoten noch eine Sequenznummer (destination sequence number) hinzu, die gebraucht wird, um die Aktualität der Routing Informationen festzustellen und um Schleifenfreiheit garantieren zu können. AODV ist ein reaktives Routing Protokoll, d.h. die Route zu einem Zielknoten wird erst dann gesucht, wenn ein Sendewunsch vorliegt. Die Weiterleitung der Pakete erfolgt hop-by-hop.

Bei der Spezifikation von AODV wurden einige vereinfachende Annahmen getroffen, zum Beispiel dass sich alle Knoten im Netz gutartig verhalten und jeder von ihnen eine, im Ad hoc Netz eindeutige, IP Adresse hat. Sämtliche AODV Nachrichten werden via UDP gesendet.

3.2 Betrieb

AODV lässt sich mit den folgenden drei Phasen charakterisieren

1. Routensuche (Route Discovery)

   Wie schon erwähnt, wird AODV erst dann aktiv, wenn bei einem Sender S ein Sendewunsch an das Ziel Z vorliegt und in der Routing Tabelle von S keine aktive Route zu Z vorhanden ist. Das ist immer dann der Fall, wenn zum Zeitpunkt des Sendewunsches in der Routing Tabelle von S keine Eintrag für Z vorliegt oder die vorhandene Route zu Z ungültig ist. Der Knoten S flutet nun eine Route Request Nachricht (RREQ) in das Netz. Der RREQ setzt sich zusammen aus einer RREQ ID, die im Zusammenhang mit der Adresse von S eindeutig ist, der aktuellen Sequenznummer und der Adresse von S sowie der destination sequence number und Adresse von Z. Ist keine destination sequence number bekannt, so muss das über entsprechende Flags im Header signalisiert werden. Alle benachbarten Knoten empfangen den RREQ und aktualisieren gegebenenfalls ihre Routing Einträge für S. Jetzt gibt es zwei Möglichkeiten:

   (a) Der Nachbarknoten N hat keine gültige Route zu Z. In diesem Fall wird N den RREQ erneut fluten. Dies wird dann solange fortgesetzt bis der RREQ den Knoten Z oder einen Knoten mit hinreichend aktuellen Routing Informationen für Z erreicht.

   (b) Der Nachbarknoten N hat eine gültige Route zu Z. Für den Fall das N≠Z wird N mit einem so genannten „gratuitous“ RREP antworten. Hierbei antwortet N stellvertretend und unentgeltlich (gratuitous) für den Zielknoten Z auf den RREQ.

   Erreicht der RREQ Z oder einen Knoten mit hinreichend aktuellen Routinginformationen so geschieht folgendes: Der entsprechende Knoten antwortet mit einer unicast Route Reply Nachricht an S. Dadurch, dass die Knoten bei der Weiterleitung des RREQ ihre

Routingeinträge für S aktualisiert haben lässt sich die so genannte Reverse Route problemlos aufbauen. Jeder Knoten, über den der RREQ gelaufen ist, kennt einen next-hop zu S und entlang der reverse route wird dann der RREP zu S geleitet. Der RREP beinhaltet sowohl die Adresse als auch die Sequenznummer von Z, die Adresse von S und eine maximale Lebensdauer für die Gültigkeit der Routinginformation.

2. Routenpflege (Route Maintenance)

   In AODV werden Routen nur solange aktuell gehalten wie sie auch aktiv benutzt werden. Erfolgt auf einer Route über eine gewisse Zeit hinweg kein Datentransfer, so werden die Einträge in den Routingtabellen für ungültig erklärt und anschließend gelöscht. Hier kann es durchaus passieren, dass eine funktionsfähige Route aus der Routingtabelle gelöscht wird, nur weil sie nicht mehr aktiv genutzt wird.

3. Routenlöschung (Route Deletion)

   Es kann nun aber auch vorkommen, das auf einer aktiven Route, d.h. es werden Daten darauf transportiert, ein Link wegbricht und somit Z nicht mehr erreichbar ist. In diesem Fall erstellt der Knoten B, dem es nun nicht mehr möglich ist seinen next-hop auf der Route zu erreichen, eine Route Error (RERR) Nachricht und sendet sie via Unicast an S. B nutzt die Reverse Route zu S um den RERR auszuliefern. Sollte es für die entsprechende Route gestattet sein, kann B auch lokal nach einer neuen Route zu Z suchen und damit die alte Route quasi reparieren. Sind diese Bestrebungen erfolglos kommt es zum Versand einer RERR Nachricht. Unmittelbar vor dem Senden eines RERR wird die destination sequence number der aktiven, nun nicht mehr verfügbaren Route inkrementiert. Eine RERR enthält ein oder mehrere Tupel der Form (Adresse,zugehörige destination sequence number), wobei Adresse die Adresse des nun nicht länger erreichbaren Knotens ist. Die genaue Anzahl an ungültigen Zieladressen wird über Headerfelder signalisiert. Erreicht der RERR S und es wird weiterhin eine Route zu Z benötigt, so startet S einen neuen RREQ mit der inkrementierten destination sequence number.

3.3 Routing Table

In Tabelle 1 befindet sich ein Überblick der benötigten Routinginformationen, die in der AODV-Routingtabelle gehalten werden müssen.

4 Secure AODV (SAODV) Routing

Um ein MANET als Ganzes gegen Angriffe zu sichern bedarf es zweierlei Schutzmechanismen: einen zum Schutz der zu übertragenden Dateneinheiten und einen weiteren zum Schutz der Routinginformationen. SAODV [Zapa05] wird für den sicheren Austausch der Routinginformationen eingesetzt. SAODV ist eine Protokollerweiterung für AODV, in der hauptsächlich Nachrichtenerweiterungen definiert werden. Im weiteren Verlauf werden wir uns auf den Bereich der Routingsicherheit beschränken.

4.1 Anforderung an einen Routing-Schutzmechanismus

Wenn man im Zusammenhang mit Netzen von Sicherheit spricht, so verbindet man dies eigentlich immer mit den folgenden Eigenschaften oder Zielsetzungen:

Feld	Erklärung
Destination IP Address	Die im Ad hoc Netz eindeutige IP Adresse des Zielknotens
Destination Sequence Number	Die letzte bekannte Sequenznummer des Zielknotens
Valid Destination Sequence Number flag	Wird benötigt um anzuzeigen ob eine gültige Squenznummer vorliegt
State and routing flags	z.B. ein valid Flag, das anzeigt, ob die Route gültig ist
Network Interface	Die Netzschnittstelle, auf der der nächste Knoten auf der Route zu erreichen ist
Hop Count	Anzahl benötigten Übertragungen, mit denen das Ziel erreicht werden kann
Next Hop	Der nächste Knoten, an den eine Nachricht weitergeleitet wird, die zum Zielknoten soll
List of Precursors	Eine Liste aller Vorgängerknoten, die diesen Knoten nutzen, um den Zielknoten zu erreichen
Lifetime	Verfalls- oder Löschzeitpunkt der Route

Tabelle 1: Inhalt der AODV Routingtabelle

4.1.1 Verfügbarkeit

Verfügbarkeit stellt die Überlebensfähigkeit des Netzes gegenüber Denial of Service (DoS) Attacken sicher. Grundsätzlich können solche Attacken auf jeder Schicht stattfinden. Auf Vermittlungsschicht sind Angriffe auf den Routingprozess denkbar, die neben weiteren Fehlfunktionen z.B. zu einer Segmentierung des Netzes und damit den Verlust der Verbindung führen können.

4.1.2 Vertraulichkeit

Vertraulichkeit ist so zu verstehen, dass bestimmte Informationen niemals an unberechtigte Stellen gelangen. Angreifer könnten aus den abgefangenen Routinginformationen Rückschlüsse auf die Topologie des Netzes ziehen. Mit diesen Kenntnissen ist es wesentlich leichter einen Angriffspunkt zu finden um das Netz maximal zu stören.

4.1.3 Integrität

Integrität garantiert, dass eine übertragene Nachricht unter keinen Umständen manipuliert wurde. Wenn es zu einer Manipulation durch Dritte gekommen ist, so kann das im Nachhinein festgestellt werden und wird im Normalfall dazu führen, dass die Nachricht verworfen wird.

4.1.4 Authentifikation

Authentifikation ermöglicht es jedem einzelnen Knoten die Identität seine Kommunikationspartners fehlerfrei und eindeutig festzustellen. Wäre dies nicht möglich, so könnte sich ein Angreifer hinter der Identität eines Dritten verstecken und daraus unerlaubte Vorteile ziehen. Darunter fallen z.B. das Erschleichen von Zugriffsrechten und das Ausspähen sicherheitsrelevanter Informationen.

4.1.5 Unabstreitbarkeit

Dem Sender können die von ihm verschickten Nachrichten eindeutig zugeordnet werden. Hat ein Knoten K eine fehlerhafte Nachricht versendet, so kann er dies im Nachhinein nicht abstreiten. Dies ist dahingehend sinnvoll, das ein Knoten E, der die fehlerhafte Nachricht von K empfangen hat, es K anlasten kann diese Nachricht gesendet zu haben. E kann nun andere Knoten davon überzeugen, dass K kompromittiert wurde.

4.2 AODV Erweiterungen

Im Bereich der Routenfindung können bei SAODV zwei alternative Nachrichtenerweiterungen zum Einsatz kommen: Zum einen die einfachere Single Signature Extension [4.2.1], zum anderen die Double Signature Extension[4.2.4]. Die Erweiterungen folgen dem in [PeBRD03] vorgegebenen Type/Length-Schema und werden jeweils direkt an die ursprünglichen Nachrichten angehängt.

4.2.1 Single Signature Extension (SSE)

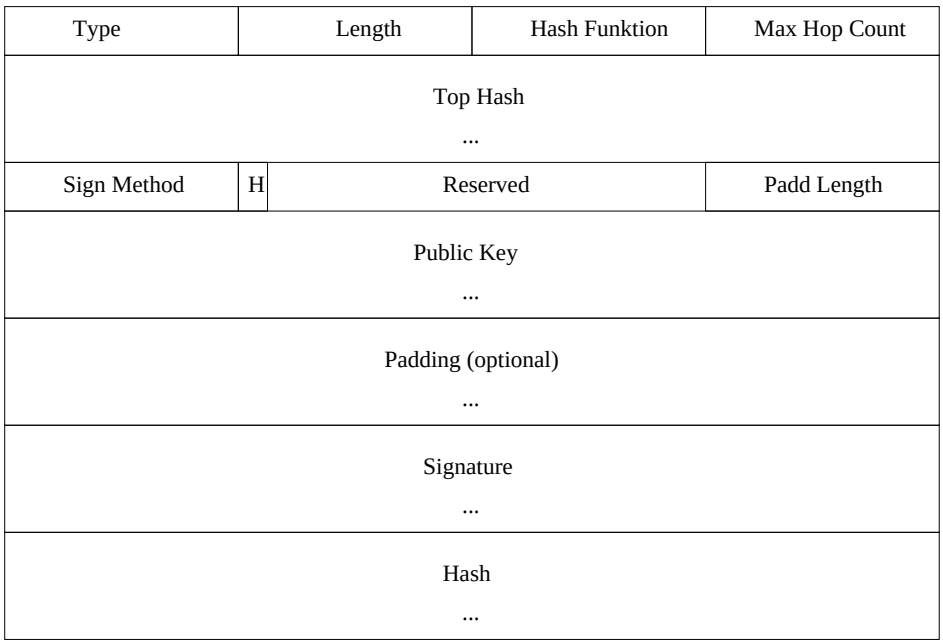

Abbildung 1: RREQ-SSE und RREP-SSE

Die Single Signature extension ist die Minimalanforderung an eine Secure AODV Implementierung. Wird ein Route Request mit der SSE Erweiterung (Abb. 1) gesendet, so ist es nur dem Zielknoten erlaubt auf den RREQ zu antworten. Sollte ein bösartiger Knoten dennoch auf eine RRQ-SSE mit einem RREP-SSE reagieren, so muss er diese signieren. Da er aber den Private Key des Zielknotens nicht kennt, muss er dafür seinen eigenen privaten Schlüssel verwenden. Durch eine in Abschnitt 4.4 genauer beschriebene Technik ist es den Knoten nun aber möglich anhand der Desination Node Address und dem Public Key festzustellen, dass die Nachricht nicht vom Zielknoten stammt. Die Nachrichtenerweiterungen für RREQ und RREP (Abb. 1) sind im Fall der Single Signature Extension identisch. Die Signatur erstreckt sich über alle vorangehenden Felder mit Ausnahme des Hop Count, der sich im ursprünglichen AODV RREQ befindet und hier nicht dargestellt ist. Der Hop Count selbst wird über eine

Hashkette gesichert. Die Bedeutung der für die Hashkette relevanten Felder Hash Function, Max Hop Count, Top Hash und Hash werden in [4.2.3] erläutert.

4.2.2 Signaturen

Signaturen werden benutzt um die Integrität und Authentizität der Nachrichten sicherzustellen. Zur Berechnung einer Signatur wir zuerst der Hash-Wert aller Felder berechnet, die vor dem Signaturfeld liegen. Zur Bildung des Hash stehen mehrere Verfahren zur Auswahl. Welches Hashverfahren für eine Signatur verwendet wurde wird über ein Flag signalisiert. Für alle standardkonformen Implementierungen ist die Unterstützung von SHA1 verbindlich. Felder, die sich von hop zu hop verändern können, werden vor der Berechnung des Hash auf Null gesetzt. Darunter fällt immer der Hop Count, bei einem RREP zusätzlich das R und A Flag im Header. Das A Flag zeigt an, dass der Versender des RREP eine Bestätigung haben möchte, wenn die Nachricht ihr Ziel erreicht hat. Bösartige Knoten könnten durch das Setzen dieses Flags dafür sorgen, das der Sender eine Empfangsbestätigung erhält, obwohl er dies nicht wünscht. Im schlimmsten Fall lässt sich daraus eine Art von DoS Attacke konstruieren. Da aber auf ein RREP maximal eine RREP-ACK gesendet wird kann dies vernachlässigt werden. Das R Flag wird für Multicastverbindungen in AODV benötigt. SAODV wurde nich entworfen um AODV Multicast zu unterstützen. Zur Verschlüsselung des Hash-Wertes stehen wieder mehrere asymmetrische Verfahren zur Auswahl, RSA ist verpflichtend. Der berechnete Hash-Wert wird vom Sender mit seinem privaten Schüssel signiert. Der signierte Hash-Wert wird anschließend als Signatur, zusammen mit dem öffentlichen Schlüssel des Senders, übertragen. Der Empfänger überprüft die Signatur, indem er den Hash-Wert der empfangenen Nachricht berechnet und diesen mit der Signatur vergleicht.

4.2.3 Hashketten

Dadurch, dass der Hop Count im AODV Header von jedem Knoten nkrementiert wird bevor er das Packet weiterleitet, lässt er sich nicht mit einer Signatur schützen. Zur Absicherung des Hop Counts werden Hashketten eingesetzt. Möchte ein Knoten S eine Routing-Nachricht versenden, so generiert er einen zufälligen Wert s (sog. seed), mit dem die Hashkette initialisiert wird. S legt einen maximalen Hop Count k fest. S wählt nun eine Hash-Funktion, die er k-mal auf s anwendet. Der daraus resultierende Wert o wir auch Top Hash genannt. Der Top Hash, die Hash-Funktion und der maximale Hop Count werden mit der Nachricht übertragen und durch die Signatur vor Veränderung geschützt. Der Sender hängt s an das Ende der Nachricht im so genannten Hash-Feld an. Er setzt den Hop Count auf null und versendet die Nachricht. Empfängt ein Knoten I eine Nachricht, so kann er nun prüfen ob der Hop Count manipuliert wurde. Der Knoten I berechnet nach dem Empfang einer Nachricht den Wert e nach folgender Vorschrift:

$$e = MaxHopCount - HopCount$$

Anschließend wendet er die angegebene Hash-Funktion e-mal auf den Wert im Hash-Feld an. Sollte das resultierende Ergebnis nicht identisch mit dem Top Hash-Wert sein, so wurde der Hop Count manipuliert. Wurde die Nachricht hingegen als authentisch erkannt und soll weitergeleitet werden, so erhöht I den Hop Count um eins. I nimmt den Wert aus dem Hash-Feld, wendet die Hash-Funktion einmalig auf diesen Wert an, und schreibt den neuen Wert an die Stelle des alten Wertes. Im Anschluss darauf wird die Nachricht weitergeleitet.

Trotzdem ist es für einen Angreifer möglich, den Hop Count zu manipulieren, ohne dass dies auffällt. Ein bösartiger Knoten erhält eine Nachricht und sendet sie weiter, ohne den Hop

Count zu erhöhen und den neuen Hash-Wert zu berechnen. Dies führt dazu, dass nachfolgende Knoten davon ausgehen, dass die Strecke zum Absender um einen Hop kürzer ist als dies in der Tat der Fall ist. Dieser Angriff wird aber erst dann zu einem Problem wenn es mehrere bösartige Knoten auf einer Route gibt. Auf eine ähnliche Art und Weise kann der Hop Count von einem bösartigen Knoten künstlich erhöht werden. Der empfangene Wert im Hash-Feld wird einfach mehr als einmal erhöht und der Hop Count dementsprechend angepasst. Solange man mit dem resultierenden Hash-Wert unterhalb des Top Hash bleibt wird auch diese Veränderung nicht erkannt. Einem bösartigen Knoten ist es aber nicht mehr möglich, durch Herabsetzen des Hop Count die Attraktivität seiner Route zu steigern, und damit Datenverkehr auf sich zu ziehen.

4.2.4 Double Signature Extension (DSE)

Szenario: Ein Knoten S_1 sucht eine Route zu Knoten Z. I ist ein zwischenliegender Knoten der den RREQ von S_1 erneut via Broadcast weiterleitet. S_2 sucht im Anschluss nach einer Route zu S_1. Auch dieser RREQ läuft über I.

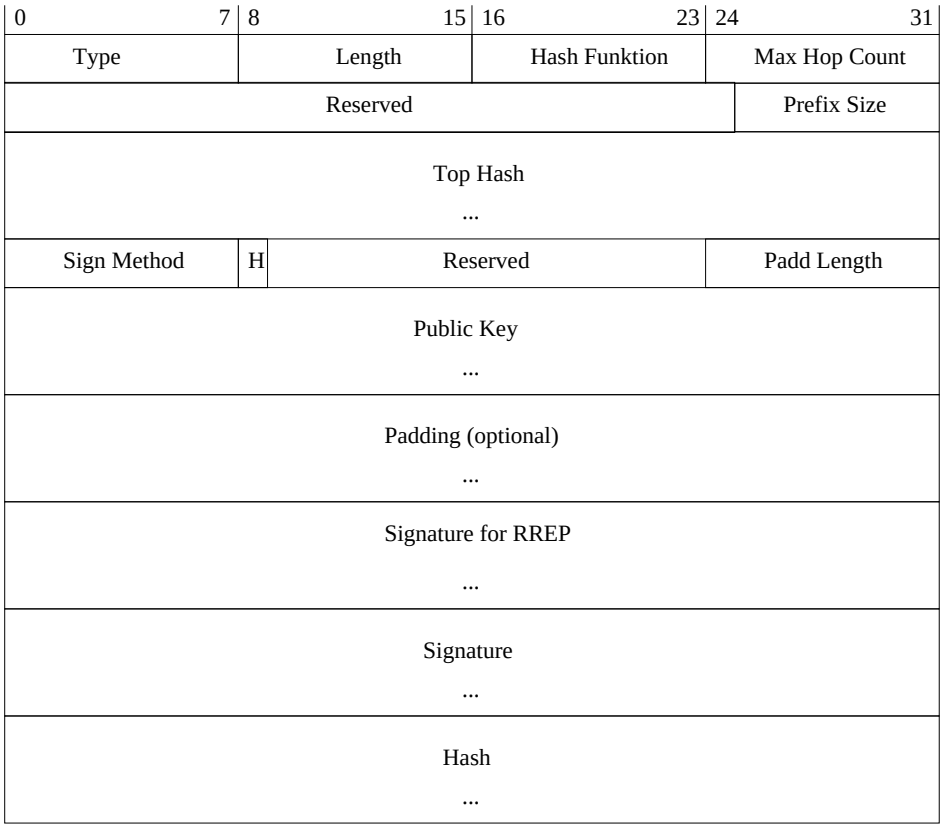

Abbildung 2: RREQ-DSEE

Der in Abbildung 2 dargestellte Double Signature Route Request unterscheidet sich vom Single Signature Route Request in Abbildung 1 nur in einem zusätzlichen Feld *Signature for RREP*. Mit dieser Signatur wird es nun wieder möglich, dass nicht nur der Zielknoten auf ein RREQ antwortet. Bei der Double Signature Extension können nun alle zwischenliegenden Knoten mit einem „gratuitous“ Route Reply antworten, wenn sie über eine „Signature for RREP“ des Destination Node verfügen. Bei diesem Feld handelt es sich um einen Art von Blanko Signatur, die S_1 mit dem Double Signature RREQ nach Z an alle Knoten verteilt. Erreicht der RREQ von S_2 nun den Knoten I, so kann dieser mit einem Double Signature Route

Reply (Abbildung 3) antworten. Aufgrund der urspünglich von S_1 stammenden „Signatur for RREP“ kann S_2 nun sicher sein, das die Routinginformation authentisch ist.

[CeGD05] gibt allerdings zu bedenken, das je nach Auslastung des Knotens entschieden werden muss, ob ein „gratuitous“ RREP erfolgen sollte. Abhängig von den in der Warteschleife befindlichen Nachrichten ist es unter Umständen besser, den RREQ weiterzuleiten und ihn nicht selbst zu beantworten.

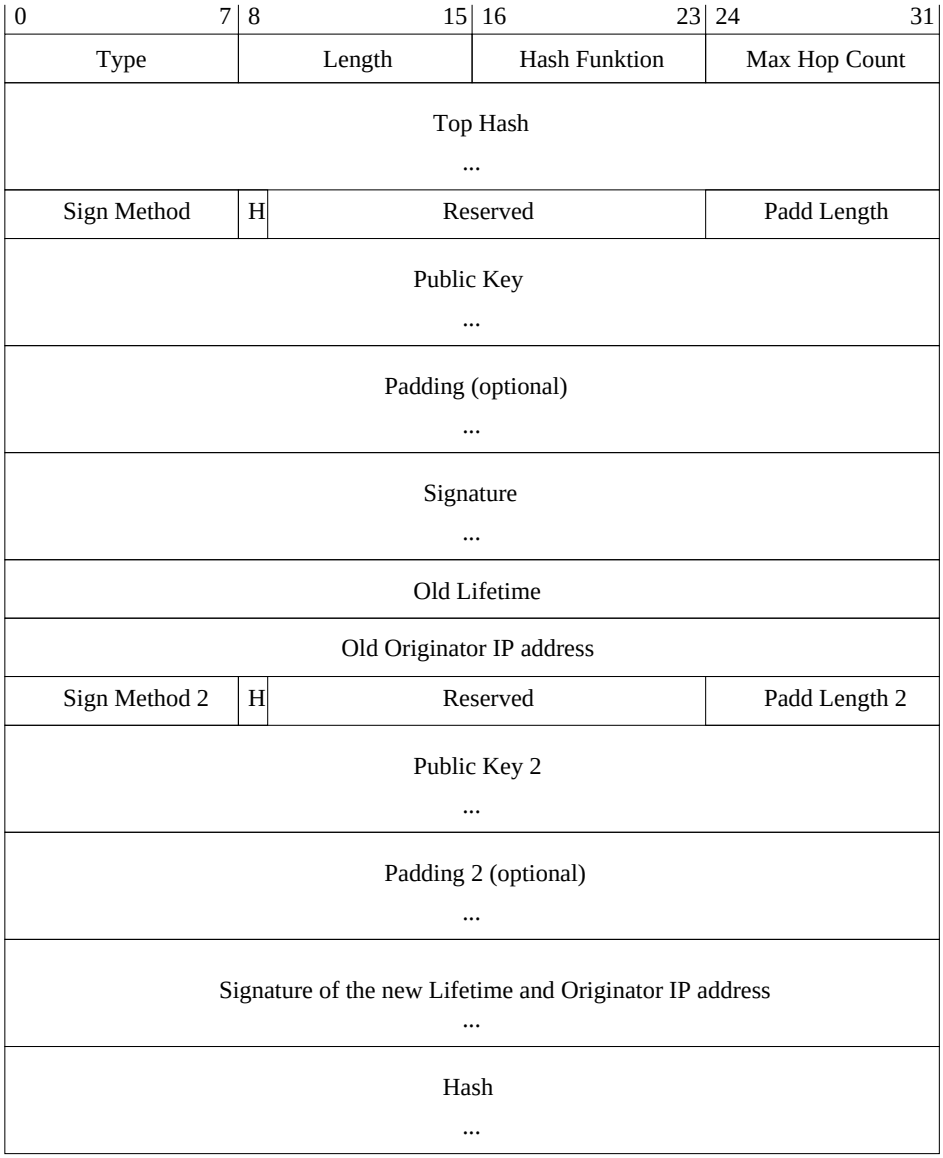

Abbildung 3: RREP-DSE

4.2.5 Weitere Erweiterungen

Der größte Teil der hier nicht im Detail vorgestellten AODV Erweiterungen beschränkt sich darauf, mittels Signaturen die Integrität und Authentizität der Nachricht sicherzustellen. Hier gibt es keine wesentlichen Neuerungen gegenüber den bereits erläuterten Mechanismen. Das Format dieser Nachrichtenerweiterungen ist unabhängig davon, ob SSE oder DSE verwendet wurden.

4.3 Betrieb

Durch die eingeführten Erweiterungen werden Anpassungen innerhalb der drei Routingphasen notwendig, die hier vorgestellt werden.

Szenario: Ein Knoten S sucht eine Route zu Knoten Z. I ist ein zwischenliegender Knoten der den RREQ von S erneut via Broadcast weiterleitet.

1. Route Discovery (Routensuche)

   Bevor S einen RREQ verschicken kann muss er entscheiden, ob er die SSE oder DSE Erweiterung benutzen möchte. Eine Nachricht sollte niemals ohne Signaturerweiterung versendet werden. S generiert einen Zufallswert s zur Initialisierung der Hashkette. Es muss ein Max Hop Count zur Berechnung des Top Hash festgelegt werden. Der Max Hop Count sollte auch in das TTL Feld des IP Headers übernommen werden. Kommt wie empfohlen ein RREQ-DSE zum Einsatz, so muss zusätzlich die Singnature for RREP berechnet werden. Dies geschieht wie folgt:

   (a) S nimmt an, er hätte einen RREQ von Z mit dem Ziel S erhalten. Was nicht der Fall ist

   (b) S reagiert auf diesen Fake-RREQ und generiert darauf einen RREP. Hierfür werden die bereits bestimmten Felder des RREQ-DSE verwendet, wie z.B. der Top Hash, und wo benötigt, die Felder des Fake RREQ.

   (c) Darauf wird mit demselben Verfahren wie im RREQ-DSE die Signatur für den RREP berechnet. Dieser Signatur ist die „Signature for RREP“ die im RREQ-DSE von S versendet wird.

   Der RREQ wird dann über das gesamte Netz geflutet.

   Empfängt I eine Nachricht und kann sie nicht beantworten, so leitet er sie unverzüglich weiter um Verzögerungen durch die Verifikation der Signatur zu vermeiden. Bevor I allerdings Änderungen an seiner Routingtabelle vornimmt, verifiziert er zuerst die Signatur. Wenn eine Nachricht nicht signiert ist oder die Signatur nicht bestätigt werden kann, so wird die Nachricht verworfen.

   Wird eine authentische RREQ Nachricht empfangen, so wird die Reverse Route zum Originator Node angelegt oder aktualisiert. Handelte es sich bei der Nachricht um einen RREQ-DSE so wird die Signature for RREP und die Adresse des Destination Node in der Routingtabelle unter dem Eintrag von S gespeichert. Ein Zwischenknoten I antwortet auf ein RREQ nur, wenn der RREQ als korrekt erkannt wurde und der Knoten in der Lage ist mit einer RREP-DSE (Abbildung 3) zu antworten. Dafür muss I die notwendigen Einträge, „Signature for RREP“, „Old Lifetime“ und „Old Originator IP address“ für den im RREQ aufgeführten Destination Node in seiner Routingtabelle zur Verfügung haben. Sollten diese Bedingungen nicht erfüllt sein, so wird er die Nachricht erneut via Broadcast weiterleiten.

   Bevor I eine Nachricht weiterleitet, muss der Hop Count inkrementiert und der neue Hash-Wert $Hash$ berechnet werden. Dabei ist h die in der Nachricht angegebene Hash-Funktion.

   $$Hash = \mathsf{h}(Hash)$$

   Erreicht der RREQ Z, so antwortet dieser nach der Verifikation der Signatur mit einem RREP-SSE.

   Empfängt ein Knoten ein RREP, erfolgt die obligatorische Verifikation der Signatur. Bei einer authentischen Nachricht wird die Route zusammen mit der Signatur, der

Lifetime und der Originator IP Address des RREP in der Routingtabelle gespeichert. Dies Informationen werden gesammelt um später mit einem „gratuitous“ Route Reply antworten zu können.

2. Route Maintenance (Routenpflege)

   Hier gibt es keine wesentlichen Neuerungen. Die in AODV verwendeten Nachrichten werden über Signaturen geschützt, die beim Empfang verifiziert werden müssen.

3. Route Deletion (Routenlöschung)

   Die Nachrichtenerweiterung für Route Error Nachrichten fügt lediglich eine Signatur hinzu. Keine Neuerung im Vergleich zu dem bisher behandelten.

   Es wird allerdings explizit darauf hingewiesen, dass die in der RERR Nachricht enthaltenen Zielsequenznummer auf keinen Fall dazu verwendet werden soll um die lokale Sequenznummer in der Routingtabelle zu aktualisieren. Die Sequenznummer wurde schließlich nicht von dem entsprechenden Destination Node heraufgesetzt, sondern von demjenigen Knoten H, der seinen Next Hop auf der Route verloren hat. H steht es frei um welche Größenordung er die Sequenznummer erhöht. Die Knoten die eine RERR Nachricht empfangen nutzen sie lediglich um zu entscheiden, ob eine Route für ungültig erklärt werden soll.

4.4 SUCV Identifiers and Addresses

Der durch Signaturen mittels Public- und Private Key Paaren erzielbare Sicherheitszuwachs fällt gering aus wenn sich der Public Key nicht eindeutig einem Knoten bzw. seiner Kennung, zuordnen lässt. Deshalb bedient SAODV sich der so genannten Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses [MoCa02], die solch eine Verbindung herstellen. SUCV Identifiers kommen aus dem Umfeld der MobileIPv6 [JoPA04] Bindingupdates. Bindingupdates werden vom „mobile node“ zum „correspondend node“ gesendet, um diesen über eine geänderte Adresse des mobile node zu informieren. Hier gilt es das „identifier ownership“-Problem zu lösen. Es muss sichergestellt sein, dass Bindingupdates nur vom mobile node durchgführt werden können und nicht von Dritten, die sich als der mobile node ausgeben. Zu diesem Zweck muss sich die Quell-IP Adresse eines Bindingupates eindeutig dem mobile node zuordnen lassen.

Bei dem Verfahren bezieht sich die Eigenschaft Statistically Unique darauf, dass es selbst bei einer grossen Menge von Teilnehmern unwahrscheinlich ist, dass es zu einer Kollision in den erzeugten Adressen kommt. Cryptographically Verifiable beschreibt den Umstand, dass es allen Teilnehmern möglich ist zu entscheiden, ob eine gegebene Adresse zu einem gegeben Public Key gehört. Umgekehrt ist es jedoch schwierig bis unmöglich aus einer gegebenen Adresse einen Public Key zu berechnen, der zu dieser Adresse passt. Es werden zwei Arten von Identifiern unterschieden:

1. *Fullidentifier*: hier wird die gesamte Länge einer Kennung für den SUCV Identifier verwendet, bei IPv6 [DeHi98] sind das 128 bit. Dabei gehen aber alle Routinginformationen verloren die dann zusätzlich in IP Packeterweiterungen mit angegeben werden müssten
2. *Halfidentifier*, hier wird lediglich die halbe Länge einer Kennung für eine SUCV Address verwendet, was es ermöglicht, eine immer noch routebare Adresse zu haben. Allerdings werden hier Kollisionen wahrscheinlicher.

Da das nachträgliche Einfügen von Routinginformationen einen zusätzlichen Overhead bedeutet sind Halfidentifier vorzuziehen. Bei einem isolierten, reinen MANET ohne Verbindung zum Internet ist auch der Einsatz von Fullidentifiern problemlos möglich.

In SAODV werden IP Adressen sehr ähnlich generiert wie SUCV Adressen. Der Hauptunterschied liegt darin, das bei SAODV der Public Key gehasht wird und nicht wie bei SUCV Adressen ein Imprint und der Public Key. Die Identifier berechnen sich wie folgt:

- SAODV_HID = SHA1HMAC_64(PublicKey, PublicKey)
- SAODV_FID = SHA1HMAC_128(PublicKey, PublicKey)

Von den beim Einsatz von SHA1 erzeugten 160 Bit Ausgaben werden jeweils nur die ersten 64 bzw 128 bit genutzt, jenachdem ob ein Full- oder Halfidentifier eingesetzt wird. Sollen es sich bei den Identifier um gültige IPv6 Adressen handeln, so gilt es noch einige Anpassungen vorzunehmen [HiDe03].

Beim Einsatz von SUCV Identifiers in IPv4 Netzen mit einer grossen Anzahl von Knoten lässt sich aufgrund der kurzen Adresslänge von 32 bit die statistische Eindeutigkeit nicht garantieren. Als gross sind in diesem Zusammenhang alle Netze anzusehen, die eine Teilnehmeranzahl von 100 übersteigen.

Durch die angesprochen Anpassungen, die gemacht werden müssen, um eine gültige IP Adresse zu erzeugen, wird die Wahrscheinlichkeit für eine Kollision erhöht. Aber auch der Fall, dass zwei Knoten einen identischen Public Key haben führt zu einer Kollision. Im Zusammenhang mit den SUCV Adressen werden in Secure AODV deshalb drei neue Nachrichten eingeführt um Adresskollisionen zu beheben. Der Ablauf dieser Kollisionsbereinigung wird im Folgenden erläutert.

Dublicated Address (DADD) Detected Nachricht

Der Knoten E empfängt eine von Knoten Z signierte Routing-Nachricht. Stellt E nun fest, das er in seiner Routingtabelle bereits einen Eintrag für einen Knoten A hat, mit derselben IP Adresse wie Knoten Z, verwirft er die Routing-Nachricht von Z. Darauf generiert E eine DADD Nachricht und schickt sie an Z. Dadurch wird Knoten Z darüber informiert, dass er eine nicht eindeutige IP-Adresse verwendet (innerhalb des Ad hoc Netzes). Damit Z diese Behauptung von E überprüfen kann, schickt E neben der IP Adresse auch den Public Key von A an Z.

New Address (NADD) Notification Nachricht

Empfängt ein Knoten Z eine DADD, oder er bemerkt selbst, dass er eine nicht eindeutige IP Adresse benutzt, so erzeugt er ein neues Schlüsselpaar. Darauf leitet Z aus seinem neuen Public Key eine IP Adresse ab. Nun informiert Z alle relevanten Knoten mittels einer unicast NADD über seine neue IP Adresse und den neuen Public Key. Eine NADD enthält beide IP Adressen (neu und alt), sowie beide Public Keys. Die NADD wir durch zwei Signaturen gesichert, jeweils erstellt mit dem alten bzw. neuen Private Key. Die „alte“ Signatur erstreckt sich über alle vorangehenden Daten, und die „neue“ Signatur sowohl über alle Daten als auch über die „alte“ Signatur.

New Address Acknowledgement (NADD-ACK) Nachricht

Knoten K, die eine NADD empfangen können diesen mit einer NADD-ACK beantworten um den Empfang zu bestätigen. In der NADD ist die neue und alte IP Adresse von Z, der Public Key von K und die Signatur von K enthalten.

Nach alle dem erzeugt Z eine Route Error Nachricht für seine alte IP Adresse. Die Verbreitung dieser Nachricht wird alle Routingeinträge für die „alte“ IP Adresse löschen, und damit die mehrdeutige Adresse.

4.5 Probleme

Wie schon im Vorfeld erwähnt sichert SAODV keinen Multicast Datenverkehr ab. Die entsprechenden Flags werden bei der Berechnung der Signaturen immer mit Null belegt. Da Multicast heute noch eine eher geringe Rolle spielt ist dies sicherlich akzeptabel.

Auf die Möglichkeit den Hop Count und damit die Distanz zwischen zwei Knoten zu manipulieren wurde eingegangen. Damit kann verhindert werden, dass eine optimale Route zwischen zwei Knoten gefunden wird. Unter der Annahme einer stark dynamischen Topologie kann es als eher unwahrscheinlich betrachtet werden, dass einzelne bösartige Knoten auf Dauer den Routingprozess global stören. Es müsste schon eine große Menge, sich koordiniert bewegender Knoten existieren, um dauerhaft das Routing zu beeinträchtigen.

Bezüglich der Sequenznummern verlangt SAODV eine Anpassung an AODV. Die einzelnen Knoten sollen in der Lage sein sich ihre Sequenznummer auch über einen Neustart hinaus zu merken. Ist dies nicht der Fall, wäre es möglich einen Knoten nach einem Neustart mit einer maximalen Sequenznummer zu impfen. Diese hätte nach der nächsten Inkrementierung je nach Implementierung unabsehbare Folgen. Aus eben diesem Grund wird auch der Destination Sequence Number in RERR Nachrichten kein Vertrauen geschenkt.

Als an Ad hoc Netzten teilnehmende Knoten sieht man heute meist Endgeräte mit begrenzter Rechenleistung und begrenzten Energiereserven. Durch die kryptographischen Berechnungen kann es dadurch zu Verzögerungen auf den einzelnen Knoten kommen. In die Wahl des entsprechenden Kryposystems sollte deshalb auch immer die Rechnerleistung der Endgeräte mit einbezogen werden. Die Belastung der Zwischensysteme ist zwar ein negativer Aspekt, er lässt sich aber mit der Zielsetzung, ein sicheres Routing zu betreiben, nicht vermeiden. Auch Zapata setzt sich mit diesem Problem in [Guer04] auseinander.

In jeder übertragenen und signierten Nachricht wird der Public Key des Sendeknotens mit übertragen. Zu Beginn ist dies sicherlich der richtige Ansatz. Hat sich ein gewisser Zustand im Netz etabliert und eine Menge von Knoten kennt den Public Key, sollte man sich diesen Overhead sparen. In der vom Standard abweichenden Implementierung von [CeGD05] wurde dies durch ein Key Absent Bit Flag umgesetzt.

Angriffe von Innen werden in SAODV nicht behandelt. Es existieren keine Mechanismen um kompromittierte Knoten zu erkennen und dann vom Routingprozess auszuschließen, z.B. über Blacklists.

5 Zusammenfassung

AODV und SAODV als eine Erweiterung von AODV lassen sich problemlos in einem Netzt parallel betreiben. Knoten, die SAODV nicht verstehen, ignorieren einfach die Nachrichtenerweiterungen. Der Ansatz von SAODV zur Sicherung der Routinginformationen ist rein proaktiv, das bedeutet, es wird im Voraus versucht Angriffe zu verhindern. Die andere Möglichkeit wäre ein reaktives Konzept, in dem während des Routings nach bösartigen Knoten gesucht und dann gegen sie vorgegangen wird. Durch den Einsatz der SUCV Adressen ist es jedem Knoten auf sichere Art und Weise möglich, die Beziehung zwischen der Adresse eines Knotens und dessen Public Key zu verifizieren. Die SUCV Addresses lösen auch ein weiteres Konfigurationsproblem: In AODV wird davon ausgegangen, dass alle Knoten eine eindeutige IP-Adresse haben. Über das Verfahren der SUCV Identifiers generieren die Knoten nun ihre eigenen, meist eindeutigen IP-Adressen. Etwaig auftretende Kollisionen werden erkannt und über die neuen Nachrichtentypen DADD, NADD und NADD-ACK beseitigt. Die Signaturen stellen die Gültigkeit der in den Routing-Nachrichten enthaltenen Informationen sicher. Den

Knoten ist es möglich zu erkennen, dass ein bösartiger Knoten falsche Nachrichten versendet. Die Sequenznummer in RREQ und RREP Nachrichten werden von SAODV über die Signatur geschützt, jede Veränderung führt dazu, dass eine Nachricht als falsch erkannt und verworfen wird. Der Hop Count ist wie zuvor beschrieben zwar immer noch angreifbar, aber bösartige Knoten haben nun nicht mehr die Möglichkeit, durch eine Verringerung den Datenverkehr auf sich zu ziehen. Black und Gray Hole Angriffe werden damit stark erschwert. SAODV ist in der Lage alle Angriffe zu handhaben, die Routinginformationen verfälschen oder modifizieren. Weiterhin lässt sich erkennen, wenn ein bösartiger Knoten versucht sich hinter einer fremden Identität zu verbergen (spoofing).

In MANETs ist es nicht möglich zentrale Infrastruktur und Komponenten ausfallsicher und erreichbar zu Verfügung zu stellen. Der Einsatz klassischer Ansätze wie beispielsweise eine Public Key Infrastruktur (PKI) mit ihren Certification Authorities ist damit ohne Anpassungen nicht möglich. Obwohl zentrale Komponenten fehlen, lässt sich mit dem Einsatz der hier vorgestellten Mechanismen ein klarer Sicherheitszuwachs innerhalb des Routingprozesses verzeichnen. Ob es jemals gelingen wird in MANETs einen vergleichbaren Sicherheitsstandard zu erreichen wie in strukturbasierten Netzen, werden zukünftige Arbeiten auf diesem Gebiet zeigen.

Literatur

[CeGD05] Davide Cerri, Alessandro Ghioni und Francesco Dolcini. A-SAODV. http://saodv.cefriel.it/, 2005.

[DeHi98] S. Deering und R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), Dezember 1998.

[EaJo01] D. Eastlake 3rd und P. Jones. US Secure Hash Algorithm 1 (SHA1). RFC 3174 (Informational), September 2001.

[Guer04] Manel Guerrero Zapata. Key Management and Delayed Verification for Ad Hoc Networks. In *In Proceedings of HiPC Workshops 2004, Electronic proceedings, Trusted Internet Workshop, paper #8, 8 pages*, December 2004.

[HiDe03] R. Hinden und S. Deering. Internet Protocol Version 6 (IPv6) Addressing Architecture. RFC 3513 (Proposed Standard), April 2003.

[JoKa03] J. Jonsson und B. Kaliski. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. RFC 3447 (Informational), Februar 2003.

[JoPA04] D. Johnson, C. Perkins und J. Arkko. Mobility Support in IPv6. RFC 3775 (Proposed Standard), Juni 2004.

[Labo] RSA Laboratories. What is the RSA cryptosystem. http://www.rsasecurity.com/rsalabs/node.asp?id=2214.

[MoCa02] G. Montenegro und C. Castelluccia. Statistically Unique and Cryptographically Veri able, 2002.

[oSTe02] National Institute of Standards und Technology. *Secure hash standard*. National Institute of Standards and Technology, Washington. Note: Federal Information Processing Standard 180-2, 2002.

[PeBRD03] C. Perkins, E. Belding-Royer und S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), Juli 2003.

[RiSA78] Ronald L. Rivest, Adi Shamir und Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM* 21(2), 1978, S. 120–126.

[Zapa05] Manel Guerrero Zapata. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. Internet Draft, September 2005. work-in-progress.

Abbildungsverzeichnis

Tabellenverzeichnis

Die Internet Indirection Infrastructure (i3): Mobilität und Overlaynetze

Tobias Schlager

Kurzfassung

Abstraktion und Indirektion spielen eine wichtige Rolle bei der Unterstützung von Mobilität, Anycast- und Multicast-Kommunikation im heutigen Internet. Aus diesem Grund empfiehlt i3 die Verwendung einer Roundez-vous-basierten Ende-zu-Ende-Kommunikation. Dadurch wird eine Entkopplung des Sende- vom Empfangsverhalten erreicht und so eine native Unterstützung verschiedenster Dienste ermöglicht. Das auf i3 aufsetzende ROAM ist eine robuste und effiziente Architektur, die speziell auf die Anforderungen für den mobilen Einsatz ausgerichtet ist. Um effizientes Routing, schnellen Handoff und Standortanonymität zu erreichen nutzt es die Möglichkeit, über die Indirektionspunkte der i3-Architektur direkten Einfluss auf das Routing nehmen zu können. Darüber hinaus ermöglicht es simultane Mobilität ohne Veränderung des TCP/IP-Protokollstacks.

1 Einleitung

Während sich das heutige festnetz gebundene Internet bereits über viele Haushalte und Unternehmen erstreckt, hat das mobile Internet das Potential, alle Bereiche des menschlichen Lebens, Arbeitens und Reisens zu umfassen. Zur vollen Entfaltung dieses Potentials sind jedoch Dienstmerkmale wie übergangslose Konnektivität und stetige Erreichbarkeit unerlässlich. Bedauerlicherweise unterstützt das heutige Internet diese Dienstmerkmale nicht, da es lediglich für die Unicast-Kommunikation stationärer Endgeräte entworfen wurde. Die Verknüpfung der IP-Adresse (als eindeutigen Identifikator von Endgeräten) mit Standortinformationen ist dabei das grösste Problem, da sich der Identifikator zur Erfüllung der obigen Merkmale bei einem Standortwechsel nicht ändern darf. Zur Lösung dieses Problems bedarf es der Erzeugung einer Indirektionsschicht, die die Standortabhängigkeit der IP-Adresse ausgleicht. i3/ROAM bietet ein solches Kommunikationsmodell, das darüber hinaus noch die Flexibilität besitzt, weitere wichtige Dienste wie Anycast- und Multicast-Kommunikation zu unterstützen.

2 Grundlagen

Die folgenden Abschnitte beschäftigen sich mit den Grundlagen dieser Arbeit. Zu diesem Zweck werden zunächst die in diesem Zusammenhang relevantesten Unterschiede zwischen mobilen Netzen und Festnetzen aufgezeigt. Anschließend bietet Abschnitt 2.2 einen Überblick darüber, wie man diese Unterschiede auf der Vermittlungsschicht auszugleichen versucht, wonach Abschnitt 2.3 den Ansatz mittels des Overlay-Konzepts auf Anwendungsschicht des ISO/OSI-Schichtenmodells näher beleuchtet. Abschließend bietet Abschnitt 2.4 eine Einführung in das *Chord*-Protokoll, das den Zugriff auf verteilte Daten ermöglicht.

2.1 Mobilität

Als Standard für die Kommunikation in drahtlosen lokalen Netzen hat sich mittlerweile der in [Schi00] näher beschriebene Standard IEEE 802.11 etabliert. Während sich im sogenannten *Managed*-Modus eine zentrale Instanz um die Kommunikation der in Übertragungsreichweite liegenden Knoten untereinander und mit der Außenwelt kümmert, erlaubt der *Ad-Hoc*-Modus die Kommunikation benachbarter Knoten ohne jegliche Infrastruktur. Da sich diese Arbeit auf Netzwerke mit zentraler Verwaltung und Festnetze konzentriert bleiben die Merkmale des Ad-Hoc-Modus im Folgenden unberücksichtigt.

Im Managed-Modus steht vor allem das *Semantic Overloading* Problem im Vordergrund. Semantic Overloading kennzeichnet die gleichzeitige Nutzung von IP-Adressen als Lokator und Knotenbezeichner. Aus diesem Grund stellt der Übergang eines mobilen Knotens in eine andere Funkzelle (Handoff) z.B. beim Wechsel des Access-Points das grösste Problem dar, da für den erzwungenen Austausch der IP-Adresse der Abbruch einer laufenden Kommunikations-Session erforderlich ist und somit ein *Seamless Handover* unmöglich wird. Unabhängig davon können durch den Einsatz von Routing-Protokollen, die auf die Anforderungen von Mobilität spezialisiert sind, noch weitere Probleme auftreten. Dazu gehören unter anderem Missbrauch der Möglichkeit zur Einflussnahme auf das Routing, Einschränkung der Verfügbarkeit des Netzes durch Angriffe gegen die Infrastruktur oder etwa der Missbrauch des geteilt genutzten Mediums zum Abhören vertraulicher Daten.

2.2 Mobile IP

Das *Mobile-IP*-Protokoll versucht das Problem des Handoffs auf Vermittlungsschicht zu lösen, indem es, um bestehende Kommunikationsverhältnisse (*Sessions*) aufrecht erhalten zu können, ein Verfahren einsetzt, das sehr stark dem Nachsende-Verfahren der Post ähnelt.

Dabei besitzt jeder *Mobile Node* (MN) einen *Home-Agent* (HA) und einen *Foreign-Agent* (FA), die, ähnlich einem Postamt, die Vermittlungsarbeit übernehmen. Bei einem Wechsel des Subnetzes teilt der Knoten seinem HA seinen neuen Aufenthaltsort mit, indem er ein *Mobility Binding* bei seinem HA etabliert. Anschließend leitet dieser alle Pakete, die für den MN bestimmt sind zum FA weiter, der sie dann an den MN zustellt. Da ein solches Mobility Binding bei jedem Wechsel der Funkzelle nötig ist, kann der FA für einen bestimmten MN häufig wechseln. Hat der MN dagegen selbst Daten zu versenden, dann schickt er diese zuerst an seinen FA, der die Pakete anschließend direkt via normalem IP-Routing an die Zielknoten versendet.

Da eine Lösung des Mobilitätsproblems auf Schicht 3 die Modifikation von Routern notwendig macht, ist aufgrund der damit verbundenen, hohen Kosten die Akzeptanz derartiger Verfahren eher gering. Zudem werden durch den Einsatz des Mobile-IP-Protokolls topologisch unkorrekte Adressen vorgetäuscht, die nicht mehr der Lokalisation eines Knotens herangezogen werden können und daher das Routing negativ beeinflussen können.

Um sich vor den in Abschnitt 2.2 genannten Sicherheitsbedrohungen zu schützen, baut Mobile IP auf eine kryptographische Authentifizierung (z.B. Keyed MD5) und eine Verschlüsselung der Daten auf der Sicherungsschicht im fremden Subnetz.

2.3 Overlays

Im Gegensatz zu den auf der Schicht 3 angesiedelten Protokollen arbeiten Overlay-Netze auf der Anwendungsschicht des Internet-Referenzmodells. Zum Versand ihrer Daten nutzen sie

die Dienste der darunter liegenden Schichten, weshalb in solchen Netzen eine völlige Transparenz der zugrunde liegenden Dienste herrscht. Hinter der Idee des Overlay-Netzes verbirgt sich der Gedanke, die Verteilung von Daten über ein virtuelles Netz, das oberhalb der eigentlichen Netztopologie angesiedelt ist, zu realisieren. Dieses enthält ausschließlich Knoten, auf denen das Protokoll des Overlay-Netzes läuft. In einem derartigen Overlay-Netz werden Ende-zu-Ende-Verbindungen durch mehrere einzelne Punkt-zu-Punkt-Verbindungen auf Transportschicht realisiert, die ihrerseits aus einer oder mehreren Schicht-3-Verbindungen bestehen. Ebenso ist das Routing völlig in das Overlay ausgelagert. Knoten, die nicht Teil des Overlay-Netzes sind, benötigen daher nur noch die Funktionalität des Unicast-Routings, um für das Overlay implizit Daten weiterzuleiten.

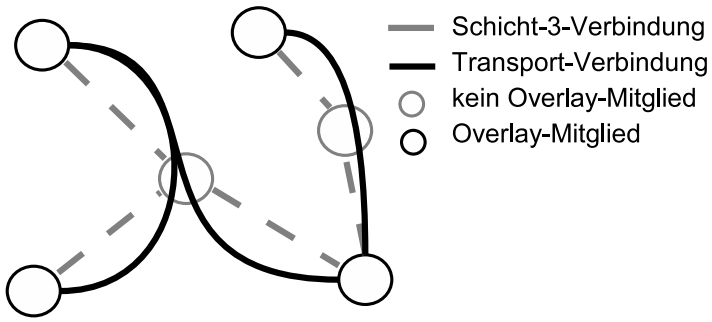

Abbildung 1: Beispiel eines Overlay-Netzwerks.

Abbildung 1 zeigt ein Overlay-Netzwerk mit Baumstruktur, ebenso können Overlays aber auch Zyklen enthalten und somit einen Mesh-Charakter besitzen. In der Abbildung sind die auf den Schicht-3-Verbindungen aufbauenden Pfade des Overlay-Baums genauso gut zu erkennen, wie das Aufspannen von Kanten mit Hilfe der Transportschicht-Verbindungen durch die Overlay-Knoten über die Nicht-Overlay-Knoten hinweg.

2.4 Chord

Beim Chord-Protokoll ([SMKK+01]) handelt es sich um ein Protokoll für den Zugriff auf verteilte Hash-Tabellen (*Distributed Hashtable, DHT*) in einer Peer-To-Peer-Umgebung. Chords einzige Funktion besteht darin, einen gegebenen Schlüssel auf den Knoten abzubilden, auf dem der Schlüssel zu finden ist. Zur Abholung und Verwendung von Daten, die eventuell mit dem Schlüssel assoziiert sind, sind separate Anwendungen nötig. Durch konsistentes Hashing (z.B. mittels der SHA-1-Funktion) werden IP-Adressen und Schlüssel in (eindeutige) m Bit lange Ganzzahlen (*Identifikatoren*) umgewandelt. Die entstehenden Identifikatoren werden in einem virtuellen Identifikator-Ring modulo 2^m angeordnet, indem ein Schlüssel k dem Knoten mit demselben Identifikator oder dem mit dem nächst grösseren Identifikator zugeordnet wird. Dieser Knoten wird dann *Successor* von k genannt.

Der eigentliche Vorteil des Chord-Protokolls besteht in seiner Skalierbarkeit bei grossen Netzen. Zum Nachschlagen eines Identifikators benötigt Chord maximal $O(log(N))$ Schritte, wenn N die Anzahl der Chord-Knoten im Netz beschreibt. Als Routing-Information verwendet Chord *Finger-Tables*, die sowohl den Chord-Identifikator eines Knotens als auch seine IP-Adresse beinhalten und maximal m Einträge besitzen. Der i-te Eintrag (*Finger*) der Tabelle enthält den Knoten, dessen Identifikator mindestens den Abstand $2^i - 1$ mit $1 <= i <= m$ (immernoch modulo 2^m) vom Identifikator des Tabelleneigners besitzt. Bei einer Anfrage mit dem Identifikator *id* schaut ein Knoten in seinem Finger-Table, ob er selbst der Successor von *id* ist, oder ob es einen Knoten in seinem Finger-Table gibt, dessen Identifikator noch näher an *id* heranreicht. Wenn dem so ist, wird die Anfrage an diesen Knoten weitergeleitet.

Ähnlich zur Intervall-Schachtelung wird bei jeder Anfrage der Abstand zu id halbiert, was in einer maximalen Suchtiefe von $log(N)$ resultiert.

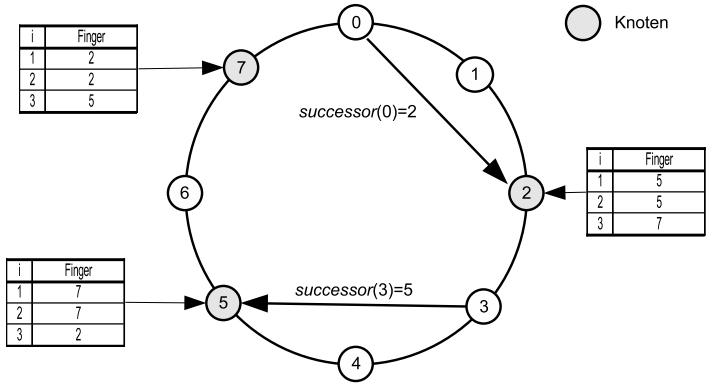

Abbildung 2: Beispiel eines logischen Identifikator-Rings.

Abbildung 2 zeigt einen Identifikator-Ring mit $m = 3$, bei dem nur die Identifikatoren 2, 5 und 7 Knoten zugewiesen sind. Die ersten beiden Einträge der Finger-Tables stimmen in diesem Beispiel überein, da der nächste Knoten auf dem Ring sowohl für den Eintrag $i = id + 2^0$ als auch für den Eintrag $i = id + 2^1$ übereinstimmt.

3 Internet Indirection Infrastructure (i3)

Die Internet Identification Infrastructure (*i3*, [SAZS+02]) ist ein Ansatz zur Vereinheitlichung und Abstraktion der Kommunikation im Internet. Ziel des Konzepts ist es, mit Hilfe einer allgemeinen Abstraktionsebene verschiedenartigste Dienste wie Multicast, Anycast und Mobilität ohne den Austausch oder die Modifikation der darunterliegenden Schichten unterstützen zu können.

Der folgende Abschnitt 3.1 wird sich zunächst mit dem Dienstmodell des Konzepts auseinandersetzen. Bevor Abschnitt 3.3 das Kapitel mit einer Betrachtung der sicherheitsrelevanten Aspekte abschließt, wird Abschnitt in 3.2 die zentrale Komponente bei der Realisierung der i3, die Trigger, erläutert.

3.1 Dienstmodell

Der Zweck der i3-Architektur besteht darin, eine Indirektionsschicht bereitzustellen, die den Sende- vom Empfangsvorgang trennen soll. Um dies zu erreichen, erlaubt i3 Pakete an logische Identifikatoren zu versenden, an denen potentielle Empfänger ihr Interesse bekunden können. Zuverlässigkeit kann nur durch separate Anwendungen erreicht werden, da im i3-Dienstmodell keine Garantien bezüglich dieses Dienstmerkmals existieren.

Obwohl das i3-Dienstmodell dem des IP-Multicast stark ähnelt, bietet das i3-Pendant zum IP-Multicast-Join deutlich mehr Flexibilität. Während einem potentiellen Empfänger beim IP-Multicast-Join lediglich die Möglichkeiten zur Verfügung stehen, Gruppenpakete zu empfangen oder diese nicht zu empfangen, ermöglicht das i3-Dienstmodell zusätzlich noch, direkten Einfluss auf das Routing der Pakete zu nehmen. Diese Flexibilität erlaubt i3 aus dem einfachen Basis-Dienstmodell heraus, Dienste wie Mobilität und Anycast-Kommunikation auf Anwendungsschicht zu unterstützen. Die Realisierung dieser Dienste durch ein einziges Overlay erspart die Erzeugung und Verwaltung redundanter Overlays und reduziert somit die

Komplexität des Netzes. Darüber hinaus kann das existierende Overlay einfach und stabil gehalten werden, da sich die Endgeräte selbst um den Aufbau effizienter Verteilbäume kümmern können.

Der Einsatz eines auf Overlays basierenden Konzepts birgt vor allem den Vorteil, dass die bisherige Struktur des TCP/IP-Protokollstacks unverändert bleiben kann und daher von der vorhandenen Infrastruktur unterstützt wird. Wie die Vergangenheit gezeigt hat, kann ein niedriger Kostenfaktor der breiten Akzeptanz eines neuen Verfahrens sehr dienlich sein. Durch das Aufsetzen weiterer Schichten steigt jedoch auch die Komplexität des Systems und stellt daher erhöhte Anforderungen an die Hardware und die benötigten Netzresourcen, was den obigen Vorteil schmälert. Ausserdem kann nicht ausgeschlossen werden, dass in manchen Fällen maßgeschneiderte Verfahren einer abstrakten Lösung überlegen sind.

3.2 i3-Trigger

Als Rendezvous-Punkt zwischen Paketen und Triggern verwendet i3 die Identifikatoren, die im Gegensatz zu IP-Adressen eine völlige Unabhängigkeit von der räumlichen Lage der Knoten garantieren. Trigger werden durch ein simples 2-Tupel dargestellt, das in seiner einfachsten Form aus einem m-Bit-Identifikator und einer IP-Adresse besteht. In ihrer einfachsten Form bestehen i3-Pakete aus einem als Zieladresse dienendem m-Bit-Identifikator *id* und einer Payload *data*, die normalerweise dem Payload-Anteil eines IP-Pakets entspricht. Potentielle Empfänger können über Trigger, die auf eindeutige i3-Knoten (*i3-Server*) abgebildet und gespeichert werden, ihr Interesse an einem Paket bekunden. Alle an einem i3-Server ankommenden Pakete, deren Identifikator mit dem eines vorhandenen Triggers übereinstimmen, werden über normales IP-Routing an die IP-Adresse des Triggers weitergeleitet.

Analog zu den Identifikatoren des Chord-Protokolls (Abschnitt 2.4) werden i3-Identifikatoren durch eine konsistente Hash-Funktion erzeugt. Gespeichert und verwaltet werden die Trigger auf speziellen i3-Knoten, die sich durch einen eindeutigen Identifikator auszeichnen und deshalb i3-Server genannt werden. Wie genau Trigger auf die i3-Server aufgeteilt und dort mit ankommenden Paket-Triggern verglichen werden wird in Abschnitt 3.2.1 eingehender erläutert werden. An diesem Punkt sei lediglich angemerkt, dass jeder Trigger auf genau einem i3-Server gespeichert und verwaltet wird.

Der obige Sachverhalt wird in Bild 3 illustriert. Die Abbildung zeigt die Realisierung eines einfachen Multicast-Dienstes mit Hilfe von Triggern. An die Stelle der Gruppenadresse tritt nun der Paket-Identifikator, über den eine beliebige Anzahl von Knoten durch das Einfügen von entsprechenden Triggern auf dem i3-Server Multicast-Daten erhalten können.

Aufgrund dieses Konzepts ist es für mobile Kommunikationspartner nicht länger erforderlich, ihre gegenseitigen Aufenthaltsorte zu kennen. Bewegt sich ein Endgerät in ein anderes Subnetz, dann muss es lediglich einen aktualisierten Trigger auf dem i3-Server, der für die Verwaltung seines Triggers zuständig ist, hinterlegen. Dies ermöglicht es, bestehende Sessions auch dann aufrechtzuerhalten, wenn beide Knoten simultan ihr Subnetz wechseln. Da auch in diesem Fall Pakete verloren gehen können, ermöglicht i3 beim Wechsel einer Funkzelle den Seamless Handover ebenso wenig wie das Mobile-IP-Protokoll. Um nicht bei jedem Sendevorgang den für denselben Identifikator zuständigen i3-Server neu ausfindig machen zu müssen, ist es möglich dessen IP-Adresse durch Server-Caching mit einer bestimmten Gültigkeitsdauer im jeweiligen Endgerät vorzuhalten. Bemerkt ein Server, dass er nicht mehr den besten Trigger für den Identifikator eines eintreffenden Pakets besitzt, setzt er bei der Weiterleitung des Pakets an den nächsten i3-Server ein *Refresh*-Flag im Kopf des i3-Pakets, das dem Server mit dem passendsten Trigger signalisiert, dass er seine IP-Adresse dem ursprünglichen Sender des Pakets mitteilen soll.

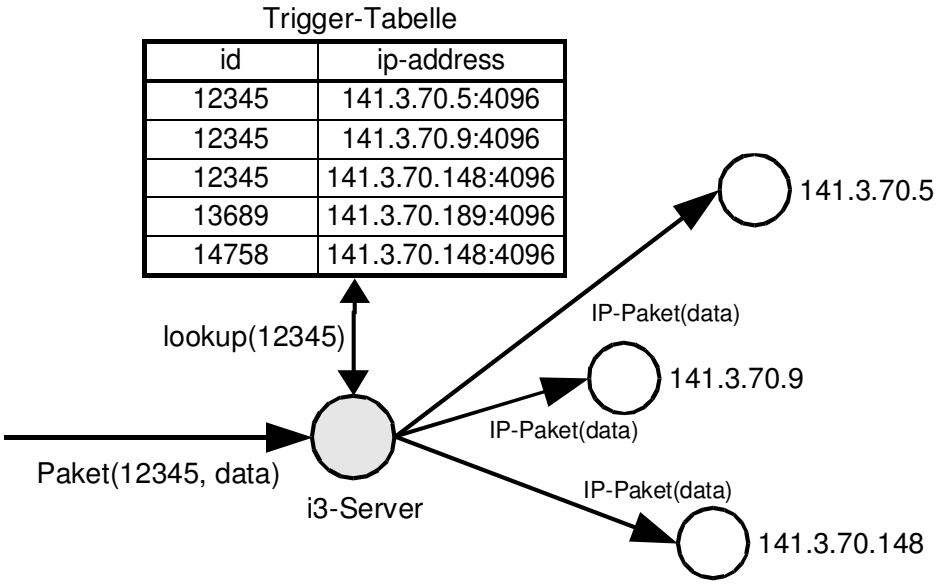

Abbildung 3: Rendezvous-basierte Kommunikation mit Identifikatoren.

Da es, wie in Bild 3 gezeigt wird, ohne weiteres möglich ist, für einen Identifikator mehrere Trigger auf einem i3-Server zu etablieren, ist ein nahtloser Wechsel von Unicast- zu Multicast-Kommunikation und wieder zurück durch das einfache Einfügen bzw. Entfernen von Triggern auf i3-Servern steuerbar. Bei grossen Multicast-Gruppen skaliert das Verfahren jedoch nicht, da die Verwaltung des Multicast-Gruppen-Triggers auf einem einzigen Server mit hoher Last verbunden ist. Für diese Fälle bietet i3 die Möglichkeit mit Hilfe der *Identifier-Stacks* (Abschnitt 3.2.2) einen Verteilbaum mit grösserer Tiefe zu konstruieren.

3.2.1 i3-Trigger-Matching

Wie bereits im vorigen Abschnitt erwähnt, bietet das exakte Matching von Triggern bzw. deren Identifikatoren die Möglichkeit, einfache Unicast- und Multicast-Dienste mit i3 zu realisieren. Darüber hinaus benötigen komplexere Dienste wie Anycast jedoch ein weitaus flexibleres Verfahren zum Vergleich zweier Identifikatoren.

Beim *Longest-Prefix-Matching* können zwei Identifikatoren auch dann zueinander passen, wenn keine exakte Übereinstimmung gegeben ist. Für m-Bit-Identifikatoren definiert i3 dazu einen *Exact-Match*-Schwellwert k mit $k < m$. Ein Identifikator id passt genau dann zu einem anderen Identifikator id_{orig}, wenn

1. id und id_{orig} ein gemeinsames Prefix von mindestens k Bits besitzen und
2. es keinen anderen Identifikator gibt, der über ein längeres gemeinsames Prefix mit id_{orig} verfügt.

Das heißt, dass der Trigger, dessen Identifikator den längsten Prefix-Match mit id_{orig} besitzt, der mindestens k Bits lang ist, zu id_{orig} passt. Bild 4 veranschaulicht dies an einem Beispiel.

Anycast-Dienste garantieren, dass ein Paket an genau einen bestimmten Knoten einer Anycast-Gruppe zugestellt wird. Als Gruppenadresse kann beim Einsatz von i3 das k Bit lange Prefix eines Identifikators verwendet werden. Wenn alle Knoten der Gruppe Trigger, die mit dem Gruppen-Prefix beginnen, registrieren, kann ein Sender den Einzelempfänger innerhalb

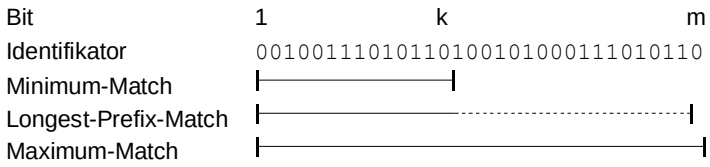

Abbildung 4: Bit-Übereinstimmung der möglichen Identifikatoren-Matches.

der Gruppe mittels eines an den Empfängerknoten angepassten Identifikators adressieren, da nur der Trigger mit dem längsten Prefix-Match als passend ermittelt wird.

Um Trigger effektiv Verwalten und Vergleichen zu können, muss sichergestellt sein, dass alle Trigger mit demselben Prefix auf einem einzigen i3-Server abgelegt sind, da sonst kein eindeutiges Longest-Prefix-Matching erstellt werden kann. i3 erreicht dies, indem ein Server für alle Trigger, deren k-Prefix[1] mit dem Identifikator des Servers übereinstimmt, zuständig ist. Dieses Verfahren bringt jedoch den Nachteil mit sich, dass sich unter Umständen eine Vielzahl sehr häufig nachgefragter Trigger auf einem einzigen i3-Server befinden können. Um solche *Hot-Spots* zu vermeiden, steht den Servern die Möglichkeit des Server-Side-Trigger-Cachings zum Load-Balancing zur Verfügung. Dabei werden häufig nachgefragte Trigger rekursiv an den Vorgänger-Server verteilt, wodurch jedoch zusätzlicher Aufwand in Folge des Versands von Refresh-Nachrichten entsteht, ohne die die gecachten Trigger nach einer bestimmten Zeit ihre Gültigkeit verlieren.

Aus dieser Art der Verwaltung von Triggern kann sich jedoch das Triangle-Routing ergeben, bei dem sich die beiden Kommunikationspartner in unmittelbarer Nähe zueinander befinden, der Rendezvous-Trigger jedoch (aufgrund des Prefix-Matching) auf einem weit entfernten i3-Server gespeichert wird. In diesem Fall werden die Pakete zunächst zum weit entfernten Server gesendet, der sie nach dem Matching der Identifikatoren an den Empfänger weiterleitet, wodurch sich eine unnötig lange Latenzzeit ergibt. Dieses Problem kann durch die Verwendung von kurzlebigen *Private Triggern* umgangen werden, die von den Kommunikationspartnern so gewählt werden, dass sie auf einem nahegelegenen Server (mit einem Pfad mit geringer Verzögerungszeit) verwaltet werden.

Zusätzlich können kürzere Latenzzeiten auch durch eine spezielle Wahl der letzten $m - k$ Bits eines Server-Triggers erreicht werden. Beispielsweise werden diese Bits bei der "Postleitzahl-Kodierung" mit der Postleitzahl des Standorts des jeweiligen Servers begonnen, so dass ein Paket mit "ähnlicher" Postleitzahl beim Matching an den räumlich nächsten Server weitergeleitet wird.

3.2.2 Identifier-Stacks

Identifier-Stacks sind Listen von Identifikatoren und werden durch ein k-Tupel der Form $(id_1, id_2, id_3, ..., id_k)$, wobei id_i entweder für eine IP-Adresse oder einen Identifikator stehen kann, repräsentiert. Pakete von der Form $p = (id_{stack}, data)$, die mit einem Identifier-Stack id_{stack} ausgestattet sind, werden über den im Stack durch die Identifikatoren-Reihenfolge festgelegten Weg weitergeleitet.

Ein mit einem Identifier-Stack versehener Trigger $t = (id, id_{stack})$ ist in der Lage Pakete nicht nur an IP-Adressen, sondern auch an weitere Identifikatoren zu versenden. Dies macht die Rendezvous-basierte Kommunikation weitaus flexibler, da hierdurch beispielsweise eine einfache Umleitung eines Pakets mit einem Identifier-Stack (id_1, id_2, id_3) durch einen Trigger

[1] Zur Erinnerung sei angemerkt, dass k die Länge des Exact-Match-Schwellwerts bezeichnet.

der Form (id_1, x, y) realisiert werden kann. In diesem Fall ergäbe sich der neue Weg des Pakets zu $x \Rightarrow y \Rightarrow id_2 \Rightarrow id_3$. Bedauerlicherweise könnten diese *hierarchischen* Trigger ausser z.B. zur Umgehung von Störungen auch zum Abhören der Kommunikation bei Man-In-The-Middle-Attacken oder für Denial-Of-Service-Attacken eingesetzt werden. Auf diese Probleme wird jedoch gesondert in Abschnitt 3.3 eingegangen werden.

Wie in Abschnitt 3.2 bereits angesprochen wurde, funktioniert das dort beschriebene Verfahren zur Realisierung des Multicast-Dienstes nicht für grosse Multicast-Gruppen, da nur ein einziger Server für die Weiterleitung der Multicast-Daten an alle Gruppen-Mitglieder verantwortlich ist. i3 löst dieses Problem durch die Kombination des obigen Verfahren mit hierarchischen Triggern. Dabei ersetzen die Gruppen-Mitglieder ihre Trigger von der Form $(id_{group}, addr_i)$ durch Ketten von Triggern $(id_{group}, x_1), (x_1, x_2), ..., (x_i, addr_i)$, bei denen die Identifikatoren (nach dem k-Prefix-Match) nicht auf den gleichen Servern gespeichert sind. Dadurch werden die Multicast-Empfänger zu Blattknoten eines Multicast-Baums, bei dem sich die Last auf die unterschiedlichen Knoten-Server verteilt.

Daneben hat die Kombination von Triggern mit Identifier-Stacks noch weitere Vorteile. Da ein nichtleerer Identifier-Stack immer an die über dem i3-Overlay angesiedelte Anwendung übergeben wird und diese dann selbst entscheiden kann, welche Information an den nächsten Identifikator im Stack weitergeleitet wird, kann sie dazu verwendet werden, Funktionalität im Netz einfach in bestehende Dienste einzubinden. Beispielsweise kann mit dieser Methode ein Video-Stream im Overlay-Netz über einen Server gesendet werden, der eine Transcodierung des Video-Materials vornimmt und es danach selbständig an die restlichen Knoten des Stacks versendet. Der Vorteil gegenüber dem bisherigen Internet besteht darin, dass auch ganze Reihen von Servern zur Datenverarbeitung mit einem einzigen Identifier-Stack adressiert werden können.

Zusätzlich können Identifier-Stacks zur Kompensation von Trigger-Verlusten eingesetzt werden, was zu einer Verbesserung der Stabilität und der Verfügbarkeit des i3-Overlays führt. Dazu fügen die Knoten zusätzliche, redundante Trigger in i3 ein, die ein Sender in einem einzigen Identifier-Stack verpacken kann. Wird der erste Trigger im Stack nicht gefunden, beispielsweise aufgrund von Serverausfällen, wird das Paket nicht verworfen, sondern an den nächsten Trigger im Stack versendet.

3.3 Sicherheit

Wie bereits in vorigen Abschnitten erwähnt, birgt die Möglichkeit mit dem Trigger-Modell direkten Einfluss auf das Routing von Paketen zu nehmen, grosse Missbrauchsrisiken.

Abschnitt 3.3.1 wird sich zunächst den Möglichkeiten für Lauschangriffen und deren Vermeidung widmen. Anschließend beschäftigt sich 3.3.2 mit dem *Trigger Hijacking*, bei dem die Gefahr von Knotenisolation besteht. Abschließend wird Abschnitt 3.3.3 die verschiedenen *Denial of Service* Bedrohungen aufzeigen und Lösungen für die einzelnen Szenarien präsentieren.

3.3.1 Lauschangriffe

In Abschnitt 3.2 wurde bereits festgestellt, dass i3 keine Unterscheidung zwischen Unicast- und Multicast-Kommunikation vornimmt. Bedauerlicherweise führt dies auch dazu, dass ein Knoten, der den Identifikator eines Unicast-Stroms kennt, diesen einfach durch das Einfügen eines entsprechenden Triggers mit seiner eigenen IP-Adresse mithören kann. Durch das i3-Dienstmodell kann weder der Sender noch der Empfänger das Abhören wahrnehmen.

Um das Abhören einer Kommunikation zu verhindern unterscheidet i3 zwischen Public und Private Triggern. Während bei Public Triggern davon auszugehen ist, dass sie aufgrund ihrer Gültigkeitsdauer nach einer bestimmten Zeit weiten Teilen des Overlay-Netzes bekannt sind, werden Private Trigger nur einmalig für die Kommunikation zweier Endgeräte gewählt und direkt nach der Beendigung der Kommunikation ungültig. Dabei sind Brute-Force-Attacken zum Erraten eines Private Triggers selbst dann statistisch aussichtslos, wenn der Identifikator so gewählt wird, dass er (wie in Abschnitt 3.2.1) auf einem räumlich nahegelegenen Server verwaltet wird, den der Angreifer kennt. Beispielsweise benötigt ein Angreifer bei einem 256 Bit langen Identifikator, bei dem der Anwendung nur 128 Bit für die Erzeugung eines zufälligen Schlüssels zur Verfügung stehen, immernoch durchschnittlich 2^{127} Versuche, um den Identifikator zu erraten.

Um die Sicherheit dieses Verfahrens zu erhöhen, schlägt [SAZS+02] vor, dass Empfänger mehrere Private Trigger besitzen und der Sender für jedes Paket einen dieser Trigger zufällig auswählt. Ein Angreifer müsste dann alle Private Trigger abhören, um die Kommunikation vollständig nachvollziehen zu können. Dies wäre jedoch mit dem Abhören auf Vermittlungsschicht oder der Übernahme des kompletten i3-Servers, der die Trigger verwaltet, identisch.

Der Austausch der Private Trigger geschieht bei diesem Verfahren über die Public-Trigger-Kommunikation. Da eine solche Kommunikation jedoch relativ leicht abzuhören ist, müssen die Private Trigger durch ein Public-Key-Kryptographie-Verfahren geschützt werden. Der Beginn einer gesicherten Kommunikation sieht bei dem obigen Verfahren demnach wie folgt aus. Knoten A verschlüsselt seinen Private Trigger mit dem Public Key von Knoten B und versendet ihn über B's Public Trigger. Knoten B entschlüsselt den Private Trigger von A, wählt seinen eigenen Private Trigger und versendet diesen über den Private Trigger von A. So kann sichergestellt werden, dass zu keiner Zeit ein Private Trigger öffentlich zugänglich übertragen wird.

3.3.2 Trigger Hijacking

Beim Trigger Hijacking kann ein bösartiger Benutzer Endgeräte isolieren oder Mitglieder einer Multicast-Gruppe entfernen, indem er deren Public Trigger löscht. Um Trigger zu löschen benötigt ein Angreifer neben dem Identifikator lediglich die IP-Adresse des Besitzerknotens, welche sich heutzutage jedoch leicht herausfinden lässt.

Zum Schutz vor Attacken dieser Art empfiehlt [SAZS+02] das Einfügen von zwei oder mehr Triggern der Form (id_S, x) und (x, ip_S). Um einen der Trigger löschen zu können benötigt ein Angreifer den Identifikator x, der aus diesem Grund geheim gewählt sein muss und dann nur S bekannt ist. Zudem sollte x noch durch die Gesichtspunkte aus Abschnitt 3.2.1 gewählt sein, so dass eine unnötige Erhöhung der Latenzzeit vermieden wird.

3.3.3 Denial of Service Angriffe

Für Denial of Service Attacken sind in i3 vor allem zwei Szenarien denkbar. Das erste Szenario richtet sich gegen beliebige Knoten im i3-Overlay. Um sie durchzuführen, fügt ein bösartiger Nutzer eine Hierarchie von Triggern in i3 ein, bei der alle Trigger am Ende der Hierarchie auf den Opfer-Knoten zeigen. Durch diese Hierarchie wird ein einzelnes, versendetes Paket beim Durchlaufen der Hierarchie vielfach repliziert. Alle erzeugten Duplikate führen bei ihrem Eintreffen dann zu einer Überlastsituation. Das zweite Bedrohungsszenario richtet sich speziell gegen die Infrastruktur des i3-Overlays. Bei dieser Attacke versucht der Angreifer durch das geschickte Einfügen einer Trigger-Hierarchie eine Routing-Schleife zu erzeugen. Jedes Paket, dass an die Wurzel dieser Schleife geschickt wird, würde bei jedem Schleifendurchlauf exponentiell repliziert werden. Abbildung 5 veranschaulicht noch einmal beide Szenarien.

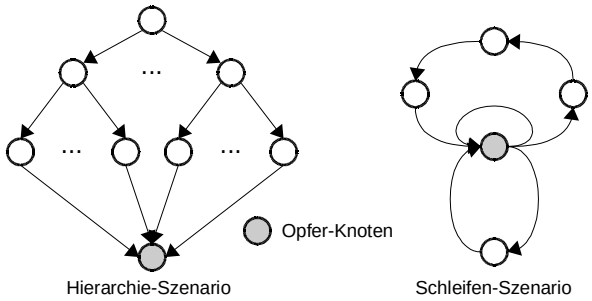

Abbildung 5: Denial of Service Angriffsszenarien.

Um die Auswirkungen solcher Attacken in Grenzen zu halten setzt i3 auf den Einsatz der folgenden drei Mechanismen.

1. i3 nimmt implizit an, dass ein Trigger der auf ein Endgerät R zeigt auch von diesem eingefügt wurde. Dies kann von einem Server sehr leicht überprüft werden, indem er R nach dem ersten Einfügen des Triggers ein Paket sendet, das eine Zufallszahl enthält. Wenn R das Paket nicht korrekt beantwortet, wird der Trigger entfernt. Da durch diesen Mechanismus die Trigger am Ende der Hierarchie entfernt werden, sobald der Server versucht hat ihre Echtheit zu verifizieren, wird der Aufbau einer kompletten Hierarchie für einen Angreifer erschwert. Trotzdem kann durch diesen Mechanismus ein Angriff durch das Einfügen einer Hierarchie nicht vollständig verhindert werden. Darauf soll jedoch im Rahmen dieser Arbeit nicht näher eingegangen werden.

2. Durch den Einsatz einer *Fair-Queueing*-Strategie wird die Auswirkung einer Routing-Schleife gedämpft, da bei dieser Strategie die Ressourcen des Servers zwischen den gespeicherten Trigger zu gleichen Teilen aufgeteilt wird. Das gibt dem Server Zeit etwaige Routing-Schleifen aufzudecken. Trotzdem kann ein Angriff mittels Routing-Schleifen grosse Erfolge erzielen, wenn ein Server nur wenige (aber eventuell stark nachgefragte) Trigger vorhält.

3. Zur Detektion von Routing-Schleifen versendet i3 nach jedem Einfügen eines Triggers, dessen Ziel keine IP-Adresse ist, ein Paket an diesen Trigger, das eine Zufallszahl enthält. Wenn dieses Paket wieder am Server ankommt, wird der Trigger entfernt. Um die Konsistenz dieser Trigger zu gewährleisten kann das Verfahren auch periodisch wiederholt werden.

4 Robust Overlay Architecture for Mobility (ROAM)

Bei der Robust Overlay Architecture for Mobility (*ROAM*, [ZLSK+03]) handelt es sich um ein über der i3-Architektur angesiedeltes Protokoll zur Bereitstellung übergangsloser Mobilität für Internet-Knoten. Aufgrund der Realisierung auf Anwendungsschicht ist es wie bei der i3-Architektur nicht erforderlich die darunterliegenden Schichten des TCP/IP-Protokollstacks zu verändern. Darüber hinaus profitiert ROAM von der Fähigkeit, mit der i3-Knoten direkten Einfluss auf das Routing und die Art der Kommunikation nehmen können und versucht dessen Vorteile für die Bereitstellung von effizientem Routing, schnellem Handover und simultaner Mobilität zu nutzen.

Zur genaueren Beleuchtung der Mechanismen gibt Abschnitt 4.1 zunächst einen allgemeinen Überblick über die Grundfunktionalitäten und Verfahren, die bei ROAM zum Einsatz kommen. Anschließend wird Abschnitt 4.2 die Dienstmerkmale des Protokolls veranschaulichen,

bevor in Abschnitt 4.3 auf die Realisierung des Linux-Prototyps an der Universität von Berkeley eingegangen wird. Abschließend wir sich Abschnitt 4.4 mit den Fragen der Sicherheit bei der Nutzung von ROAM beschäftigen.

4.1 Dienstmodell

Im wesentlichen bietet ROAM Anwendungen eine auf den mobilen Einsatz hin optimierte Schnittstelle zur Nutzung von Diensten wie Unicast-, Multicast- oder Anycast-Kommunikation. Dabei folgt ihr Dienstmodell grundsätzlich dem aus Abschnitt 3.1 bekannten Dienstmodell von i3.

Im Gegensatz zu den Adressen des Mobile-IP-Protokolls bieten Trigger höhere Flexibilität, da sie neben Endgeräten auch Sessions oder Personen bezeichnen können. Diese Flexibilität kann beispielsweise für den Schutz der Anonymität der Knotenstandorte oder die Umleitung der Kommunikation einer bestimmten Person an ein für sie verfügbares Gerät verwendet werden. Ein weiterer Unterschied besteht in der Fehlertoleranz der beiden Verfahren. Während der Ausfall des Home-Agents oder seines Netzes zumindest für die Funktion von Mobile IP in Version 4 fatale Folgen hätte, kann dies in ROAM durch den Einsatz zusätzlicher Trigger ausgeglichen werden. Schnelle Handoffs sind in Mobile IP mit hohem Aufwand verbunden und daher von entsprechender Komplexität, die in den Protokollstack eingebunden ist. ROAM dagegen bietet durch die Möglichkeit der Auslagerung dieser Mechanismen in Anwendungen eine einfache und stabile Protokollinstanz.

4.2 Dienstmerkmale

Nachdem im vorigen Abschnitt das Dienstmodell von ROAM erläutert wurde, werden im folgenden die Eigenschaften des Protokolls vorgestellt. Dazu wird sich Abschnitt 4.2.1 zunächst mit den Mechanismen für ein effizientes Routing beschäftigen, bevor Abschnitt 4.2.2 das Verfahren beim Wechsel einer Funkzelle untersucht. Abschließend erklärt Abschnitt 4.2.3 wie ROAM die Anonymität des Standorts eines Knotens schützt.

4.2.1 Effizientes Routing

Ein von ROAM bereitgestellter Routing-Pfad auf Overlay-Ebene ist nie höher als $O(log(n))$. Die kumulierte Verzögerung an jedem Hop führt jedoch trotzdem zwangsläufig auf eine höhere Latenzzeit als beim Einsatz von Mobile IP, bei dem die Pakete bei ihrer Weiterleitung nur die untersten Schichten des TCP/IP-Protokollstacks durchlaufen müssen. Um Pakete direkt über weite Strecken über IP versenden zu können, verwendet ROAM die Mechanismen des *Trigger-Server-*, des *Mobility-Aware Trigger*-Cachings und des *Trigger-Samplings*.

Während beim Trigger-Server-Caching die Adressen häufig benötigter Server vorgehalten werden, um Pakete direkt via IP-Routing an diesen weiterleiten zu können, gilt das Trigger-Sampling der Vermeidung unnötig hoher Latenzzeiten durch das Overlay-Routing über weit entfernte Server. Dazu legt ein Knoten eine Server-Heuristik durch Messung der Round-Trip-Zeit beim Versand von Paketen mit zufällig gewählten Triggern an. Nachdem die Heuristik (nach jedem Standortwechsel) angelegt ist, kann der Knoten den Server mit der niedrigsten Round-Trip-Zeit verwenden.

Beim Mobility-Aware Trigger-Caching werden die beiden oben beschriebenen Verfahren mit der Idee kombiniert, dass Knoten häufig Standortwechsel in räumlich nahe Gebiete vollziehen, während Wechsel in weit entfernte Gebiete nur selten vorgenommen werden. Um diesem Verhalten gerecht zu werden, werden im Cache nur noch Einträge vorgehalten, die für die

Orte, die ein Knoten erfahrungsgemäß (wenn auch nur selten) besucht, optimal sind. Bei jedem Subnetzwechsel erstellt der mobile Knoten eine Server-Heuristik nach obigem Verfahren und vergleicht diese mit den Einträgen in seinem Cache. Wenn der neu gefundene Trigger ein sehr viel niedrigere Verzögerung aufweist als der Trigger mit dem bisher niedrigsten Wert, ist dies ein Anzeichen dafür, dass sich der Knoten in einem Gebiet befindet, dass er bisher noch nicht besucht hat. In diesem Fall wird, wenn der Cache voll ist, der älteste nicht benutzte Trigger mit dem neuen Trigger ersetzt. Führt der Einsatz des neuen Triggers dagegen nicht zu einer so dramatischen Verbesserung wird im Falle eines vollen Caches der bisher nächste Trigger mit dem neuen ersetzt.

4.2.2 Effizienter Handoff

Um die Paketverluste während eines Funkzellenwechsel möglichst niedrig zu halten verwendet ROAM die Verfahren des *Fast-* und des *Multicast-based Soft*-Handoff.

Zur Minimierung von Paketverlusten werden beim Fast-Handoff die Trigger so gewählt, dass sie auf räumlich möglichst nahegelegenen Servern oder Servern mit geringer Latenzzeit verwaltet werden. Dies führt dazu, dass die Zeit, in der der Server ankommende Pakete an die nicht mehr verwendete Adresse des alten Triggers weiterleitet, minimiert wird.

Beim Multicast-based Soft-Handoff fügt der Knoten, der den Subnetz-Wechsel durchführt, einen Trigger mit demselben Identifikator und seiner neuen IP-Adresse ein, sobald er im neuen Subnetz eine IP-Adresse (z.B. über DHCP) erhalten hat. Damit werden die nachfolgenden Pakete an beide IP-Adressen weitergeleitet und der Knoten kann über die momentan beste, verfügbare Verbindung empfangen. Leider ist diese Methode nur dann anwendbar, wenn der Knoten auch in der Lage ist, auf zwei Netzwerkschnittstellen gleichzeitig zu lauschen.

4.2.3 Location Privacy

Durch das schon öfters angesprochene Verfahren zur Verringerung der Verzögerungszeiten durch Einfügen speziell ausgewählter Trigger auf nahegelegenen Servern entsteht das Problem, dass die Anonymität des Standorts eines Knotens nicht gewährleistet ist, sobald der Standort des Verwaltungsservers bekannt ist.

Um die Anonymität eines Knotens zu gewährleisten, ohne dass sich die Verzögerungszeit der Übertragung unangemessen erhöht, kann der Trigger umgekehrt auch so gewählt werden, dass er räumlich nahe des Kommunikationspartners verwaltet wird. Zusätzlich kann die Dauer eines eventuellen Handoffs niedrig gehalten werden, indem ein weiterer Trigger eingefügt wird, der wiederum nahe des anonymen Knotens verwaltet wird.

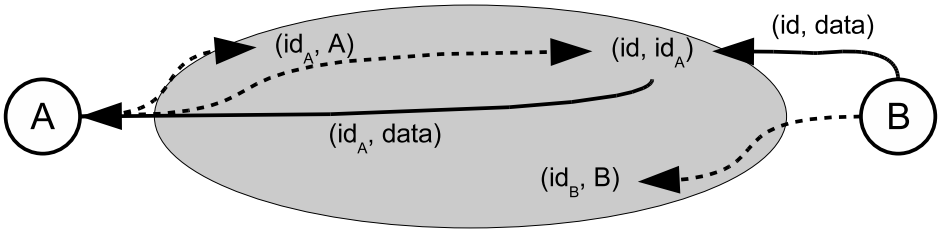

Abbildung 6: Wahrung der Standortanonymität von ROAM-Knoten.

Abbildung 6 veranschaulicht dieses Konzept. In diesem Fall wird der Standort des Knotens A durch das Einfügen der Trigger (id_A, A) und (id, id_A) vor dem Knoten B verschleiert.

4.3 Realisierung

ROAM wurde an der Universität von Berkeley als User-Level-Proxy unter der Verwendung der virtuellen Schnittstelle *TUN*, die standardmäßig in alle Linux-Betriebssysteme mit Kernel 2.4 oder höher integriert ist, implementiert. Der Proxy sorgt sowohl für die Umsetzung von i3-Paketen in Transportpakete und umgekehrt als auch für das Einfügen und Aktualisieren der Trigger. Die TUN-Schnittstelle ermöglicht es, Pakete von User-Level-Anwendungen an andere User-Level-Anwendungen zu versenden und von diesen Daten zu empfangen. Um zu vermeiden, dass Pakete von Anwendungen beim Senden nach der Bearbeitung im ROAM-Proxy wieder über die TUN-Schnittstelle zurück an die Anwendung geleitet werden, nutzt die Software *iptables*-Regeln, die das Forwarding von Reply-Paketen und DNS-Anfragen direkt an die IP Routing-Tabelle ermöglichen.

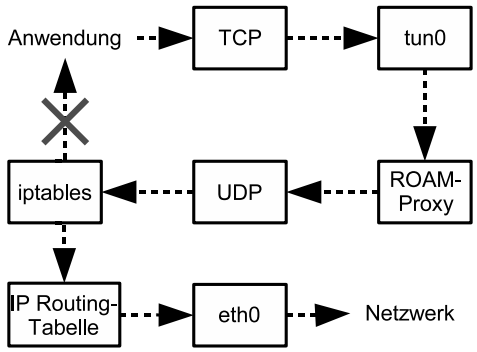

Abbildung 7: Weg des Datenversands beim Einsatz des ROAM-Proxies.

Abbildung 7 zeigt Ablauf beim Senden eines Pakets von der Userland-Anwendung aus. Beim Empfangen wird derselbe Weg in entgegengesetzter Richtung beschritten.

Der wohl grösste Vorteil des Proxy-Konzepts ist, dass die Anwendungen keine Kenntnis von der Existenz von ROAM besitzen müssen und daher unverändert funktionieren können.

4.4 Sicherheit

Das grösste Sicherheitsproblem, vor deren Lösung ROAM steht, ist der Lauschangriff. Wie schon in Abschnitt 3.3.1 angesprochen, ermöglicht das Konzept zur Vereinfachung der Multicast-Kommunikation das Abhören einer Kommunikations-Session bei Kenntnis des dazugehörigen Identifikators durch einfaches Einfügen des Triggers mit eben diesem Identifikator.

In [ZLSK+03] wird dazu die aus Abschnitt 3.3.1 bekannte Kombination aus Private Triggern und Public-Key-Kryptographie angeführt. Diese birgt jedoch den gravierenden Nachteil, dass für ihr Funktionieren eine Public-Key-Infrastruktur vorausgesetzt werden muss. Public Keys könnten natürlich auch mit DNS verwaltet werden, in diesem Fall wäre ROAM jedoch nur so sicher wie der jeweilige DNS.

Darüber hinaus ist auch die Sicherung der Public-Trigger durch ein EXCLUSIVE_ID-Flag im Header des Triggers, das andere Knoten daran hindert Trigger mit demselben Identifikator einzufügen, denkbar. Für die Multicast-Kommunikation muss dann jedoch wieder auf die Private Trigger ausgewichen werden, da nur diese als sicher angenommen werden können und daher keine zusätzliche Sicherung benötigen. Dennoch ist auch dieser Ansatz angreifbar. Ein potentieller Angreifer könnte beispielsweise darauf warten, dass die Aktualisierung eines Triggers durch den Besitzerknoten fehlschlägt. In diesem Fall kann er die Kommunikations-Session durch Einfügen eines Triggers mit seiner eigenen IP-Adresse übernehmen. Eine solche Attacke ist wiederum nur durch kryptographische Verfahren zu verhindern.

5 Bewertung

ROAM wurde bereits an der Universität von Berkeley getestet und mit Mobile IP verglichen. Simuliert wurde in einer Session-basierten Umgebung zum Test der Erzeugung, Verwaltung und Routen-Messung von IP-, Mobile-IP- und i3-Netzwerken zum Einsatz. Genaueres zu den eingesetzten Protokolleinstellungen und -erweiterungen sind in [ZLSK+03] zu finden. Zum Vergleich werden die beiden Mobile IP Routing-Strategien *Bidirectional Tunneling* (*bi*) und *Triangular Routing* (*tri*) herangezogen.

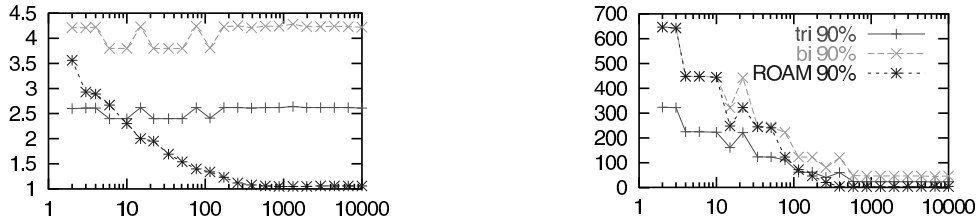

Abbildung 8: IP Stretch in Abhängigkeit zur Anzahl der H/F-Agents bzw. der i3-Server bei zufälligem (links) und Heimnetz-nahem (rechts) Bewegungsmodell.

Die Simulationen haben gezeigt, dass Mobile IP in den Fällen, dass sich der Mobile Node oder sein Kommunikationspartner mit hoher Wahrscheinlichkeit in geringer Entfernung zu seinem jeweiligen Heimnetz aufhält, von einer wachsenden Zahl zur Verfügung gestellter Home Agents profitiert. Zum einen befindet sich in diesem Fall entweder der Mobile oder der Correspondent Host mit hoher Wahrscheinlichkeit innerhalb des Heimnetzes und zum anderen wird durch eine wachsende Anzahl an Home- bzw. Foreign-Agents die Möglichkeit für das Auffinden eines nahegelegenen Agents erhöht, der das Triangel-Routing-Problem minimiert. Auch ROAM profitiert in diesen Szenarien von einer Erhöhung der verfügbaren i3-Server, so dass der IP Stretch[2] beider Protokolle auf ihren jeweiligen Kommunikationspfaden annähernd gleich sind. Während die Verzögerungszeit bei durchweg zufälligen Bewegungsmodellen beim Mobile-IP-Protokoll nahezu konstant bleibt, auch wenn die Anzahl der verfügbaren Home Agents gesteigert wird, profitiert ROAM bei diesen Modellen in hohem Maße von einer Erhöhung der verfügbaren i3-Server, da durch Mechanismen wie Trigger-Server-Caching weiterhin Pfade mit weit besseren Verzögerungen erzeugt werden können.

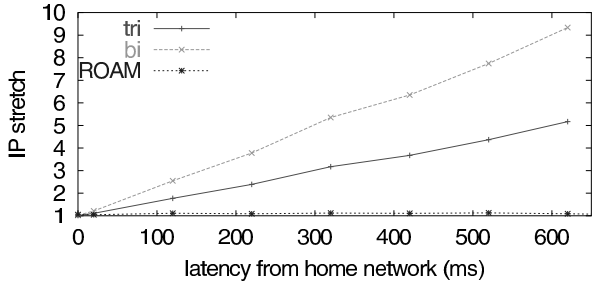

Abbildung 9: IP Stretch in Abhängigkeit zur Entfernung vom Heimnetz.

Die oben beschriebenen Ergebnisse bestätigen sich in einem Szenario mit zufälligem Bewegungsmodell und fester Server-/Home-Agent-Anzahl (Abbildung 9). Während die Verzögerungszeit bei Mobile IP mit Zunahme der Entfernung der Kommunikationspartner von ihrem jeweiligen Heimnetz linear steigt, bleibt die Verzögerung bei ROAM nahezu konstant.

[2]Mit *IP Stretch* wird das Verhältnis der gemessenen Verzögerung zur optimalen Verzögerung auf dem Pfad bei direktem IP Routing bezeichnet.

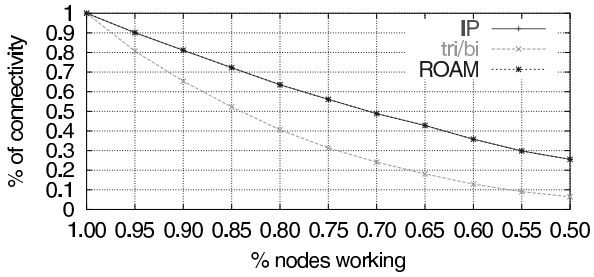

Abbildung 10: Konnektivität in Abhängigkeit zur Ausfallwahrscheinlichkeit eines Knotens.

Darüber hinaus wurde eine Betrachtung der Fehlertoleranz (Abbildung 10) vorgenommen, bei der man die Fähigkeit bei gegebener Wahrscheinlichkeit für einen Knotenausfall zwischen zwei mobilen Knoten eine Verbindung erzeugen zu können gemessen hat. Diese Messung hat ergeben, dass die Stabilität von ROAM der von IP nahe kommt, wohingegen beim Einsatz von Mobile IP mit einem starken Abfall der Konnektivität zu rechnen ist.

Abschließend wurde in [ZLSK+03] eine experimentelle Betrachtung der Handoff-Mechanismen von ROAM durchgeführt. Beim Multicast-based Soft Handoff entstehen keinerlei Paketverluste. Jedoch disqualifiziert er sich für einen Vergleich mit Mobile IP, da er die Existenz zweier Netzwerk-Schnittstellen voraussetzt. Beim Cold-Switch, bei dem ein klassischer IP-Adress-Wechsel vorgenommen werden muss schneiden beide Protokolle wieder ähnlich ab, da die Anzahl der verlorenen Daten mit der Dauer der Trennung korrespondiert. Beim Einsatz von ROAM kann diese Zeit jedoch durch die Verwendung eines räumlich nahegelegenen Servers verkürzt werden.

Literatur

[SAZS+02] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker und Sonesh Surana. *Internet Indirection Infrastructure*. Proceedings of the ACM SIGCOMM. August 2002.

[Schi00] Jochen Schiller. *Mobilkommunikation: Techniken für das allgegenwärtige Internet*. Addison Wesley. 2000.

[SMKK+01] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek und Hari Balakrishnan. *Chord: A Scalable Peertopeer Lookup Service for Internet Applications*. Proceedings of the ACM SIGCOMM. 2001.

[ZLSK+03] Shelley Zhuang, Kevin Lai, Ion Stoica, Randy Katz und Scott Shenker. *Host Mobility Using an Internet Indirection Infrastructure*. Proceedings of the First International Conference on Mobile Systems, Applications, and Services (MobiSys 2003). 2003.

Abbildungsverzeichnis

Die Vereinigung zweier Rivalen: HIP+i3=Hi3

Daniel Pathmaperuma

Kurzfassung

Das heute Internet ist für die einvernehmliche Kommunikation zwischen festen Knoten geschaffen worden. Inzwischen haben sich die Anforderungen jedoch derart weiterentwickelt, dass ihnen mit den Möglichkeiten des herkömmlichen IP nicht mehr ohne weiteres bzw. gar nicht entsprochen werden kann. Heutzutage ist ein effizienter Schutz vor Denial-of-Service Angriffen ebenso wünschenswert (und vielleicht sogar erforderlich) wie die Möglichkeit, eine Verbindung auch dann aufrecht zu erhalten, wenn sich ein Endhost innerhalb der Netztopologie bewegt. Eine Extremvariante stellt die gleichzeitige Bewegung beider Kommunikationspartner dar. Erweiterungen von IP in Form von Overlay-Protokollen stellen eine Möglichkeit dar, dieses Problem anzugehen. In dieser Arbeit werden die Protokolle i3, insbesondere secure-i3 und HIP vorgestellt. Vorteile und Schwachstellen werden angesprochen und verglichen. Die Idee der Kombination von i3 und HIP zu Hi3 wird erklärt und deren Bewertung angeführt.

1 Einleitung

Es gibt verschiedene Ansätze, die Probleme, die IP in heutigen Anwendungen aufwirft, anzugehen. Zwei Ansätze sind i3/secure-i3 und HIP. Im Folgenden werden beide Ansätze kurz erklärt und deren Stärken und Schwachstellen näher betrachtet. Danach wird auf die Idee der Vereinigung dieser beiden Ansätze eingegangen.

2 i3

Die Grundidee von i3 besteht darin, das Senden vom Empfangen zu entkoppeln (vergleiche [SAZS+02]). Dabei schickt der Sender Pakete von der Art *(id, data)* an das Netzwerk. Der Empfänger bekundet sein Interesse am Empfang von Datenpaketen, indem er so genannte *Trigger* von der Form *(id, address)* setzt.

Empfängt das i3 Netzwerk ein Paket mit einer bestimmten ID, so leitet es das zugehörige Datenpaket an all jene Empfänger weiter, die einen Trigger für diese ID gesetzt haben. Der Empfänger kann durch das Setzen und Löschen von Triggern selbst entscheiden, wessen Datenströme er empfangen will und wessen nicht. Bemerkenswert ist hier, dass es sich durchaus nicht nur um einen einzelnen Empfänger handeln muß, es kann auch eine ganze Gruppe von Empfängern geben. So wird beispielsweise bei Multicast-Anwendungen die Netzwerklast vom Sender auf das Netz verlagert.

Die Trigger sind bei i3 256 Bit lang, die in Präfix und Suffix (je 128 Bit) geteilt sind. Das Präfix eines Triggers bestimmt gleichzeitig den i3-Node, auf dem er gespeichert wird. Somit kann durch die Wahl des Triggers bestimmt werden, wo diese gespeichert werden, der Nutzer hat hier also (im Gegensatz zu IP) die Möglichkeit, Einfluss auf die Route zu nehmen.

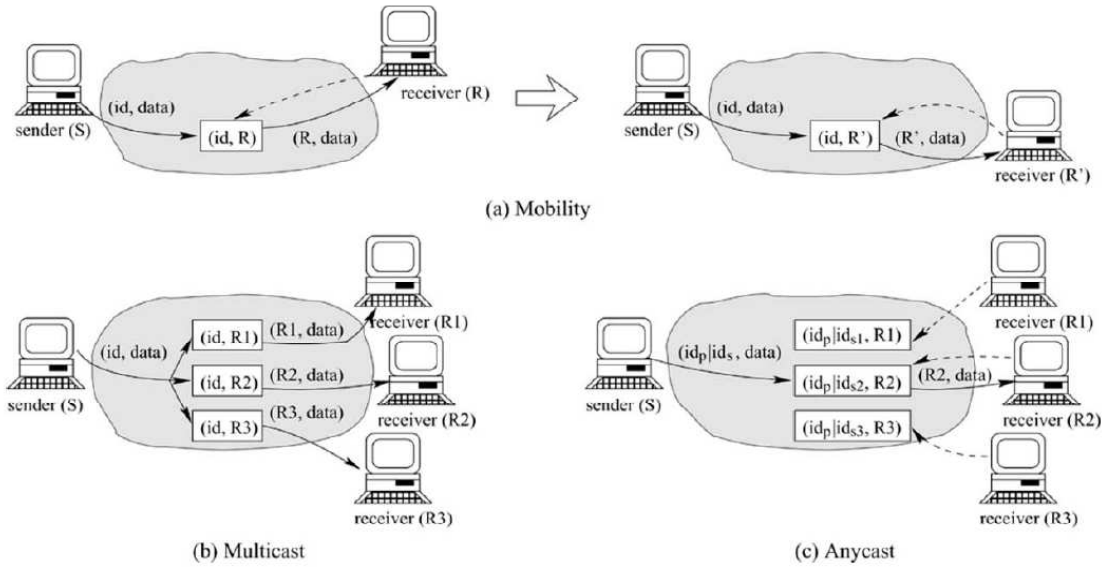

Abbildung 1: i3 bringt die Features *mobility*, *multicast* und *anycast* von Haus aus mit

Neben diesem *Multicast* bringt i3 noch weitere nützliche Features mit, die mit reinem IP so nicht ohne weiteres möglich sind.

Dies ist zum einen die Möglichkeit der Kommunikation zwischen mobilen Endgeräten. Ein Sender muß die Zieladresse eines Datenpakets nicht kennen. Vielmehr kann der Empfänger eigenständig seine Trigger setzen und aktualisieren, wenn er seinen Standort bzw. seine Adresse wechselt. Die benutzte „Adresse“ ist hier nicht länger die tatsächliche IP-Adresse sondern lediglich die ID, die Sender und Empfänger untereinander aushandeln. Der Sender muß von diesem Wechsel nichts bemerken und kann einfach weiterhin an die ihm bekannte ID senden.

Schließlich besteht noch die Möglichkeit des *Anycast*. Hier schickt der Sender ein Datenpaket an eine Gruppe von Empfängern. Es soll jedoch nicht jedes Gruppenmitglied dieses Datenpaket empfangen sondern *genau einer* aus der Gruppe. Auch dies lässt sich mit i3 realisieren.

Es ist an dieser Stelle anzumerken, dass bei i3 zwei Hosts nie direkt miteinander kommunizieren sondern Nachrichten stets an das i3 Netzwerk übergeben und von diesem in Empfang nehmen.

2.1 Probleme von i3

Ein großes Problem von i3 ist seine Anfälligkeit gegenüber Denial-of-Service Angriffen. Viel schwerwiegender ist allerdings die Tatsache, dass i3 aufgrund seiner großen Freiheiten beim Einfügen von Trigger-IDs einem Benutzer die Möglichkeit gibt, unter zu Hilfenahme der Netzwerk-Infrastruktur seinen eigenen Anfragen zu exponenzieren und dann auf ein Opfer umzulenken.

Bei einem DoS Angriff wird der Server von einer großen Zahl von Anfragen regelrecht „überschwemmt“, so dass er unter der Last zusammenbricht und seinen Dienst nicht mehr erfüllen kann. Es gibt verschiedene Varianten eines DoS Angriffs, manche zielen darauf ab, durch möglichst datenintensive Zugriffe die Bandbreite des Angriffsopfers auszulasten, andere gehen gezielt gegen einzelne Dienste auf dem Server vor und versuchen diese durch das Senden fehlerhafter Pakete lahmzulegen. Hinzu kommt eine mögliche Ausnutzung bekannter Implementierungsfehler der verwendeten Software.

Eine weitere Variante stellt ein *Distributet Denial of Service (DDoS)* Angriff dar. Die meisten „interessanten“ Angriffsopfer sind sehr robust (sie sind ja darauf ausgelegt, eine große Zahl von Anfragen zu bearbeiten). Deshalb arbeiten bei einem DDoS Angriff viele Clients zusammen, da sie mit ihren kombinierten Ressourcen das Angriffsopfer überbieten können. Bei einem solchen Angriff können im Prinzip beliebig viele Angreifer zusammenarbeiten und so die Last beliebig steigern. Dabei müssen gar nicht immer alle beteiligten Angreifer von dem Angriff wissen oder gar damit einverstanden sein, oft wird das Angriffsprogramm als Wurm über das Internet verbreitet und führt den Angriff dann ohne Wissen und Zutun des Besitzers des betreffenden Rechners aus.

Die betroffenen Hosts haben keine Chance, sich diesem Angriff auf IP-Ebene zu entziehen. Die einzige Möglichkeit wäre ein Adress-Wechsel, was jedoch in den meisten Fällen nicht in Frage kommt, da dann nicht nur der Angriff ins Leere gehen würde, sondern auch jede „sinnvolle“ Dienstanfrage.

Bei den Triggern wird zwischen öffentlichen (public) und privaten (privat) Triggern unterschieden. Die ID eines öffentlichen Triggers ist allgemein bekannt und wird ähnlich den IP-Adressen/Domain-Namen über DNS oder über einen öffentlichen Look-Up Service propagiert. Dies ist notwendig, da ja sonst niemand einen Serverdienst in Anspruch nehmen könnte, der dessen Trigger-ID noch nicht kennt. Über diese öffentlichen Trigger wird jedoch nicht der Dienstverkehr abgewickelt, sie dienen lediglich dazu, private Trigger-IDs auszuhandeln. Über diese privaten Trigger-IDs (die für jeden Client neu und eindeutig sind) wird der eigentliche Verkehr abgehandelt. Während die privaten Trigger jedoch jederzeit gelöscht werden können, ist dies mit den öffentlichen Triggern nicht ohne weiteres möglich, da der Server sonst seine Erreichbarkeit einbüßen würde. Tatsächlich hat ein Server in einem i3 Netz also nur sein IP-Adresse gegen seinen öffentlichen Trigger als Angriffsadresse eingetauscht.

In i3 ist also ein, im Gegensatz zu IP, geringfügiger Schutz gegen DoS Angriffe enthalten. Ein angegriffener Server kann sogar Zeitweise seine öffentlichen Schlüssel löschen, um einem Angriff auszuweichen. Das würde zwar dazu führen, dass er für neue Verbindungen nicht mehr zu Verfügung steht, alle bestehenden Verbindungen währen hiervon jedoch nicht betroffen, da sie ja über einzelne private Trigger laufen.

Dagegen gibt es für potentielle Angreifer noch die Möglichkeit auf der IP-Ebene einen Angriff auf den *i3-Node (Knoten)* durchzuführen, auf dem die öffentlichen Trigger des eigentlichen Angriffsopfers gespeichert sind. So bleibt der eigentliche Server zwar vom Angriff verschont, kann jedoch auch nicht mehr erreicht werden, da seine öffentlichen Trigger nicht mehr zugänglich sind.

Dies ist möglich, da ja i3 nur ein Overlay-Netz ist und somit letztendlich die einzelnen Hosts untereinander via IP kommunizieren. Durch geschicktes Einfügen eines Triggers, der auf den Angreifer zeigt, kann der Angreifer die IP des i3-Nodes in Erfahrung bringen, auf dem dieser Trigger gespeichert ist. Da bei i3 Trigger mit gleichem Präfix auf den gleichen Servern gespeichert werden, muß der Angreifer seinen Trigger nur so wählen, dass er das gleiche Präfix benutzt, wie sein Opfer. Nun kann er einen regulären DoS Angriff auf den i3-Node fahren und so verhindern, dass andere die öffentlichen Trigger des Opfers erfahren können, das Opfer ist somit nicht erreichbar.

Das Hauptproblem von i3 liegt jedoch nicht in seiner Anfälligkeit gegenüber DoS Angriffen, sonder darin, dass es einem potentiellen Angreifer die Mittel dazu praktisch frei Haus liefert, da jeder i3 Nutzer Trigger in das Netzwerk einfügen kann, und diese Einfügeoperationen keinerlei Kontrollen unterliegen.

So kann ein Angreifer beispielsweise ohne weiteres den Datenverkehr zwischen zwei ihm bekannten Hosts belauschen, indem er einfach einen weiteren Trigger einfügt und den betreffenden Datenstrom als Kopie an sich selbst weiterleitet (vgl. Bild 2a).

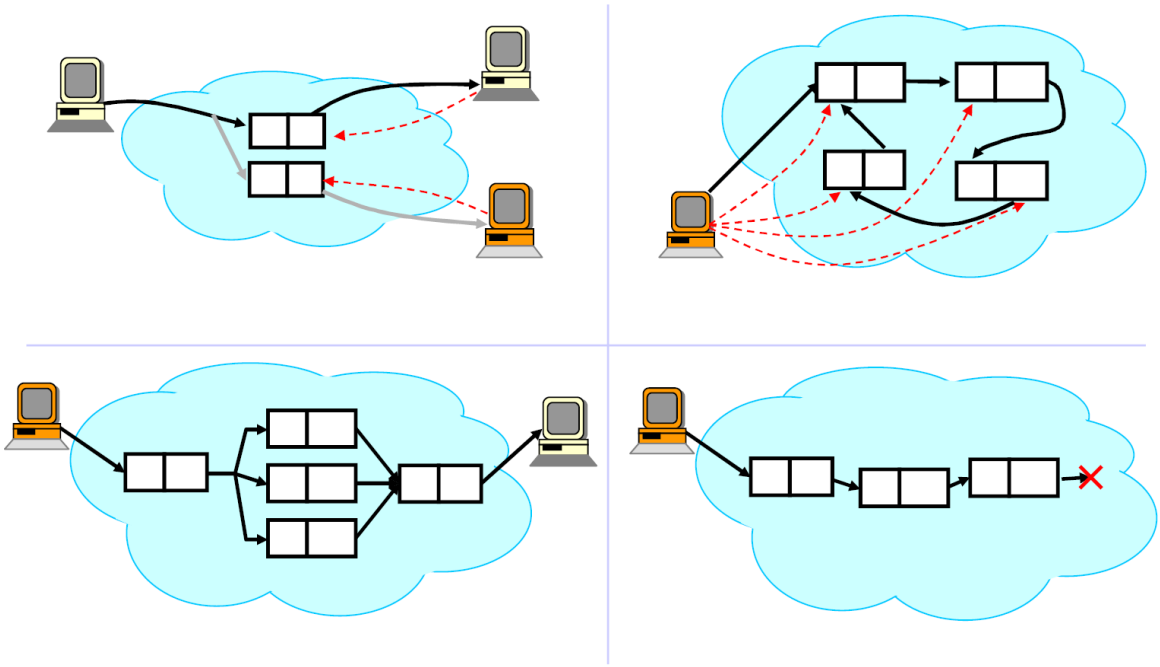

Abbildung 2: reguläres i3 erlaubt Trigger-Strukturen, die nicht wünschenswert sind

In Bild 2b wird gezeigt, wie ein Angreifer das Netzwerk an sich angreifen kann, indem er einen Ring von Triggern bildet. Datenpakete würden so von einem i3-Node zum nächsten weitergeleitet, ohne ein Ziel zu finden. Auf diese Weise wird viel Netzwerk-Kapazität nutzlos verschwendet. Ähnlich ist auch der Ansatz, der in Bild 2d veranschaulicht wird. Hier werden Pakete bewußt in eine Sackgasse geschickt, an deren Ende sie gelöscht werden. Auch dies kostet Netzwerk-Ressourcen.

Das gefährlichste Konstrukt zeigt jedoch Bild 2c. Hier spannt der Angreifer zunächst einen beliebig großen Baum aus Triggern auf. Am Ende verweist jedes Blatt auf einen Host bzw. Trigger. So kann man also unter Nutzung des i3 Netzes auch als einzelner Angreifer mit einer sehr schmalen Netzanbindung einen DDoS Angriff starten, da jedes gesendete Datenpaket vervielfacht wird, bevor es dem Empfänger zugestellt wird.

3 secure-i3

Da das herkömmliche i3 Protokoll noch einige Schwachstellen bezüglich *Denial-of-Service* Angriffen und der freien Triggerwahl hatte, wurde es in [ALPS03] an der University of California, Berkeley zu *secure-i3* weiterentwickelt.

3.1 Verbesserungen zu i3

Im Internet stellen *Denial-of-Service (kurz DoS)* Angriffe eine Bedrohung für alle angeschlossenen Hosts dar. Besonders „lohnenswert“ sind derartige Angriffe jeodch vor allem auf beliebte/bekannte/berühmte Server (z.B. spiegel.de, microsoft.com oder, erst vor wenigen Tagen, nsa.gov).

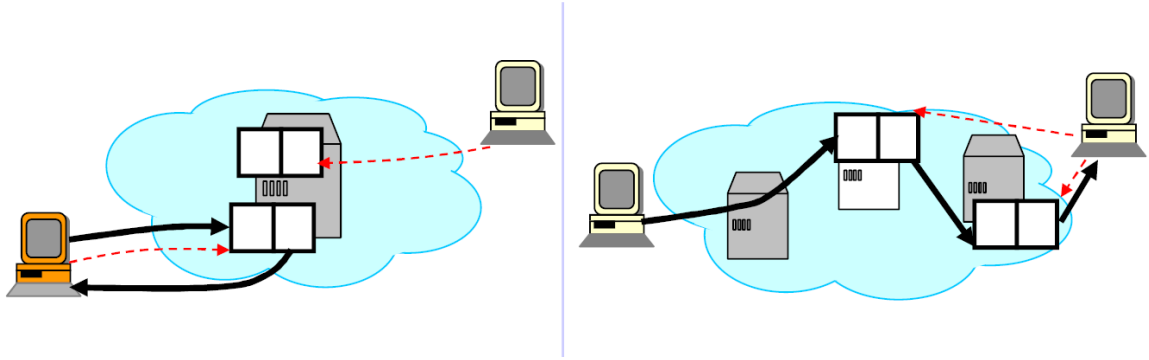

Abbildung 3: die IP-Adresse wird versteckt, indem „wichtige“ secure-i3 Nodes nie direkt mit Endknoten kommunizieren

Wie bereits in 2.1 erwähnt, kann ein DoS Angriff auf IP Ebene das i3 Netz so treffen, dass ein Server nicht mehr erreicht werden kann auch wenn er gar nicht direkt betroffen ist. Secure-i3 erweitert i3 nun dahingehend, dass ein DoS Angriff auch indirekt nicht länger möglich ist.

Eine wesentliche Änderung zum regulären i3 besteht darin, dass bei den secure-i3-Nodes zwischen Endknoten und Zwischenknoten unterschieden wird. Um zu vermeiden, dass i3-Nodes, die z.B. die öffentlichen Trigger eines Servers speichern, direkt angegriffen werden können, kommunizieren diese im secure-i3 niemals direkt mit Endhosts, sondern nur mit anderen i3-Nodes. So wird vermieden, dass die (wichtigen) i3-Nodes, die die öffentlichen Trigger speichern, Ziel eines DoS-Angriffs werden.

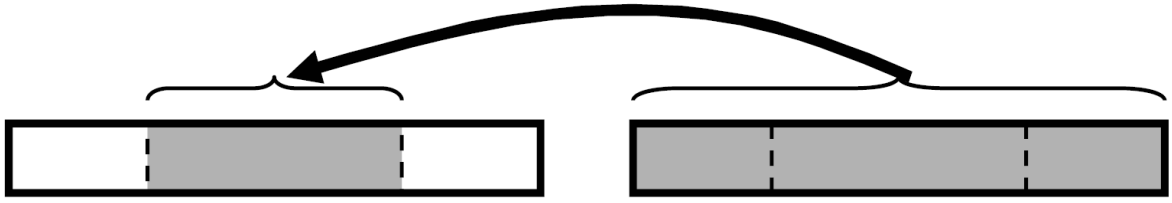

Abbildung 4: die 128 Schlüsselbits von x werden aus den 256 Bits von y berechnet

Außerdem wurden die 256-Bit Trigger des original-i3s in drei Felder (64-Bit Präfix, 128-Bit Schlüssel und 64-Bit Suffix) unterteilt. Über „Constraints“ (Beschränkungen) wird vermieden, dass neue Trigger beliebig gesetzt werden können, es können nun nur noch zyklenfreie Baumstrukturen entstehen, Rückkopplungen sind nicht mehr möglich.

Ein Trigger von der Form (x,y) wird über sogenannte *left-Constraints/l-Constraints* derart beschränkt, dass x nur noch in Abhängigkeit von y gewählt werden kann. Dazu wird der 128-Bit Schlüsselteil von x über eine Hashfunktion aus den 256 Bit von y berechnet. Unter der Annahme, dass die Hashfunktion kryptographisch „sauber“ implementiert ist (das heißt solange es zu keinen Kollisionen der verwendeten Hash-Werte kommt), ist es nun nicht mehr möglich, von einer bekannten Adresse zu einer gewünschten Adresse weiterzuleiten. Dies schließt die Bildung von Zyklen (vgl. Bild 2b+c) ebenso aus, wie das Abhören von Datenströmen, die von einer bekannten Adresse kommen (näheres hierzu in 2.1).

Fasst man die bisher erwähnten Maßnahmen zusammen, so ist ein Angriff auf IP-Ebene weder direkt auf einen Host (IP-Adresse versteckt) noch auf den i3-Node (keine direkte Kommunikation zwischen Endhosts und public Trigger Server), der die öffentlichen Trigger eines Hosts speichert mehr möglich (vgl. Bild 3).

Ein Angriff auf i3-Ebene ist jedoch noch immer möglich! Ein Angreifer kann natürlich ganz regulär bei einem Server einen privaten Trigger beantragen und einen DoS Angriff auf diesen

ausführen. Hier hat der Angegriffene nun jedoch die einfache Möglichkeit, diesen privaten Trigger einfach wieder zu löschen, der Angriff läuft dann ins Leere.

Diese Methode ist zwar sehr effizient bei Angriffen über private Trigger (sehr unrealistischen Szenario), hat aber unerwünschte Nebenwirkungen wenn der Angriff auf einen öffentlichen Trigger ausgeführt wird. Dann nämlich führt ein Löschen dazu, dass neue Hosts den betroffenen Server nicht mehr erreichen können. Bemerkenswert ist hierbei jedoch, dass all jene Clients, die bereits einen privaten Trigger mit dem Server ausgehandelt haben (im Gegensatz zu IP) ungestört weiter mit diesem kommunizieren können.

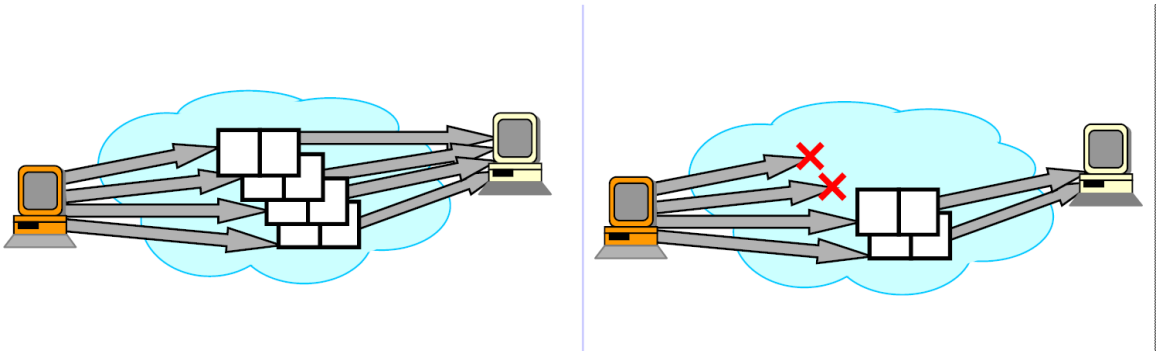

Abbildung 5: durch die Reduzierung der Anzahl öffentlicher Trigger kann ein DoS Angriff abgeschwächt werden

Um zu verhindern, dass im Falle eines Angriffs auf die öffentlichen Trigger der Server komplett unerreichbar wird, hat jeder Server eine größere Zahl öffentlicher Trigger. Ein Client sucht sich daraus zufällig einen aus. Wird nun einer dieser Trigger angegriffen, so kann er gelöscht werden. Dadurch erhöht sich die durchschnittliche Zeit, die ein Client braucht, um einen Verbindung aufzubauen, dafür sinkt aber auch die Serverlast.

Schließlich besteht noch die Möglichkeit einen DoS-Filter-Server zwischenzuschalten. Dieser leitet im Normalfall Pakete einfach weiter. Vermutet er jedoch einen DoS-Angriff, so verlangt er vor der Weiterleitung vom Client die Lösung eines „Puzzles“. Diese Puzzle können kleine kryptographische Probleme sein, die sich ohne Aufwand überprüfen lassen, deren Berechnung jedoch eine gewisse Zeit in Anspruch nimmt (eine Sekunde wäre für einen normalen Nutzer vermutlich eine akzeptable Zeit, mit einer Frequenz von einer Anfrage pro Sekunde erzeugt man jedoch keine erdrückende Serverlast).

Zur Entfernung von Sackgassen wird ein *pushback* Technik verwendet. Stellt ein i3-Node fest, dass ein Paket nicht weitergeleitet werden kann, so schickt er eine Nachricht an den Absender-Node. So können nicht mehr benötigte Trigger schnell entfernt werden.

3.2 Infrastruktur

Das i3 Protokoll stellt an die zu benutzende Infrastruktur gewisse Anforderungen. Zum einen wird vorrausgesetzt, dass ein einzelner i3-Node sicher und richtig funktioniert. Weiterhin wird vorrausgesetzt, dass die Verbindung eines Hosts zum ersten i3-Node sicher ist, so dass ein Angreifer nicht mithören kann. Außerdem sollte das Belauschen des Verkehrs zwischen einzelnen i3-Nodes nicht möglich oder zumindest sehr schwierig sein (wie es auch bei heutigen IP-Routern der Fall ist). Dies schließt ein Belauschen auf i3-Ebene jedoch nicht aus.

Schließlich sind i3-Nodes Teil der Netzwerk-Infrastruktur, so dass man eine deutlich höhere Kapazität und Lastenrobustheit von ihnen erwarten kann, als von normalen Endknoten.

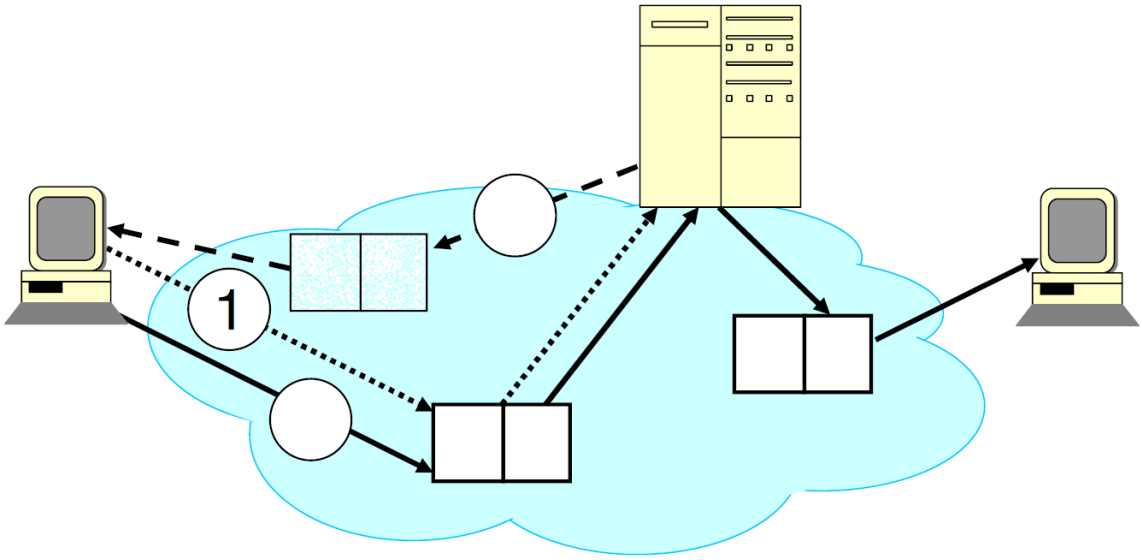

Abbildung 6: ein DoS Filter Server schützt den Endhost vor einem DoS Angriff, indem er z.B. Challanges an die anfragenden Hosts stellt

3.3 Mehraufwand und Einschränkungen

Natürlich sind die genannten Maßnahmen mit einem gewissen Mehraufwand innerhalb der secure-i3-Nodes gegenüber regulären i3-Routern verbunden. Im Folgenden wird der Mehraufwand an den einzelnen eingefügten Stufen abgeschätzt. Außerdem werden Einschränkungen der Service-Vielfalt gegenüber i3 angesprochen.

Zunächst ist festzustellen, dass die Neuerungen von secure-i3 gegenüber herkömmlichem i3 hauptsächlich den Kontroll-Pfad betreffen. Der Datenpfad bleibt unberührt und ist somit ebenso effizient wie bei i3.

Beim Einfügen von neuen Triggern entsteht gegenüber i3 ein gewisser Mehraufwand, da der einzufügende Trigger mittels der erwähnten Constraints geprüft werden muß. Außerdem muß eine invertierte Tabelle angelegt werden, die eine Rückverfolgung des Pfades für ein *pushback* im Falle eine Sackgasse verwendet werden kann. Hierbei ist anzumerken, dass dieser Mehraufwand nur beim ersten Einfügen anfällt, für eine spätere Aktualisierung ist er nicht mehr notwendig. Eine detaillierte Aufwandsanalyse [ALPS03] hat ergeben, dass ein vollständiges (erfolgreiches) Einfügen in etwa doppelt so lange dauert wie bei i3. Jedoch ist zu bemerken, dass ein Misserfolg (z.B. im Falle eines fehlgeschlagenen Constraint-Checks) deutlich weniger Zeit beansprucht (etwa 1/3 - 1/2) als eine Einfügeoperation in i3.

Auch der Mehraufwand durch die Kombination von öffentlichen und privaten Triggern fällt nur einmalig beim Verbindungsaufbau an. Danach läuft die Kommunikation über die ausgehandelten privaten Trigger ebenso effizient wie bei i3.

Auch die Möglichkeit, Prä- und Suffix von Triggern selbst zu wählen bleibt größtenteils erhalten, einzige Einschränkung ist die Kürzung von 128 auf 64 Bit. Eine Einschränkung die sich aus dem Verbot für beliebige Trigger ergibt, ist, dass eine Empfängerseitige Service-Komposition nicht mehr ohne weiteres möglich ist, da ein Einfügen eines Triggers von einer festen ID zu ein festen ID nicht mehr erlaubt ist. Dies kann jedoch umgangen werden, indem der Empfänger den Sender einfach bittet, Pakete nicht mehr direkt an ihn sonder über einen Dritten zu schicken. Dies wird in [ALPS03] als akzeptabel angesehen.

3.4 praktische Überlegungen

Das hier vorgestellte Protokoll hat nur dann einen Sinn, wenn man der Integrität der einzelnen secure-i3-Nodes auch trauen kann. Dies wäre der Fall, wenn die Infrastruktur von einem einzigen Anbieter zur Verfügung gestellt wird. Sobald es (wie im heutigen Internet) eine Vielzahl von Infrastrukturanbietern gibt, muß man sich Gedanken über das gegenseitige Vertrauen der Anbieter untereinander machen. Hier kann ein einzelnes „schwarzes Schaf“ die Sicherheit des ganzen Netzes beeinträchtigen.

Weiterhin ist auch die beste Möglichkeit, öffentliche Trigger zu propagieren noch unklar. Denkbar wäre z.B. DNS für die Verteilung öffentlicher Trigger-IDs zu benutzen, eine andere Möglichkeit wäre es, ein Look-Up-Protokoll (z.b. *Chord*) zu nutzen. Die einfachste aller Möglichkeiten wäre wohl ein Hashwert des Namens bzw. der Adresse. Hier muß jedoch noch genau geprüft werden, welche Methode ein Maximum an Sicherheit, Robustheit und Erreichbarkeit sicherstellt.

4 HIP

Während bei IP im allgemeinen ein Host mit seiner IP-Adresse gleichgesetzt wird, trennt das *Host Identity Protocol (HIP)* die Identifizierung von der Adressierung. Dazu hat jeder Host sein eigenes *Host Identity Tag (HIT)*. Dieses besteht aus dem 128 Bit langen Hash-Wert über den öffentlichen Schlüssel des Hosts. Die Adressierung erfolgt nun anhand dieser *HITs*, die je nach Situation dynamisch der aktuellen IP-Adresse des betreffenden Hosts zugeordnet werden.

4.1 HIP Base Exchange

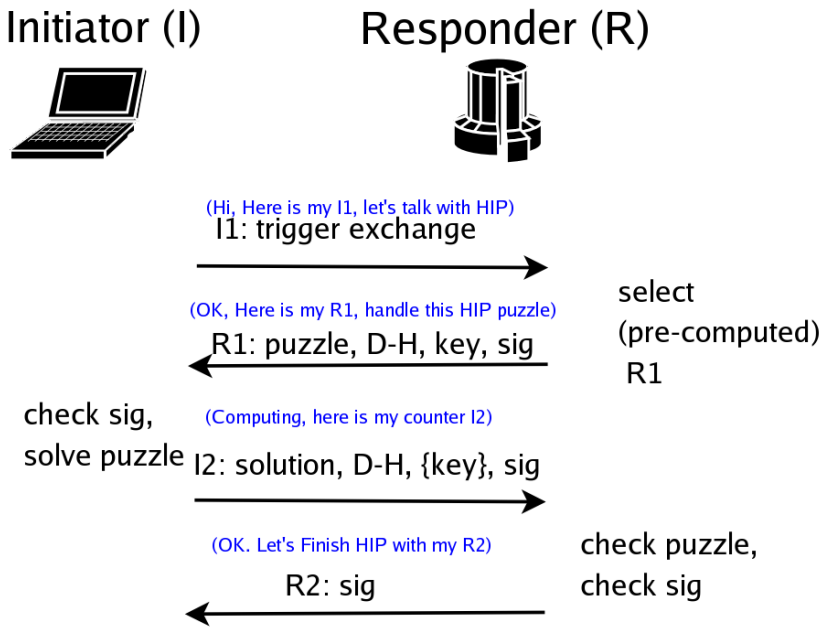

Abbildung 7: Nachrichten beim Aufbau einer HIP-Verbindung

Der Verbindungsaufbau wird *HIP Base Exchange* genannt. Dabei wird über einen Diffie-Hellman-Schlüsselaustausch ein gemeinsamer Schlüssel generiert und die Authentizität beider

Kommunikationspartner überprüft. Mit dem gewonnenen Schlüssel kann nun eine sichere IPsec-Verbindung aufgebaut werden, über die die Nutzdaten gesicher fließen.

Der HIP Base Exchange läuft dabei folgendermaßen ab: Zunächst sendet der Verbindungsinitiator I ein *I1* genannte Nachricht. Sie enthält hauptsächlich die HIT der Initiators und informiert den Empfänger R über den Verbindungswunsch. Dieser antwortet darauf mit der Nachricht *R1*. Sie enthält neben der ersten Hälfte des Diffie-Hellman Schlüsseltauschs und dem public-Key auch ein kryptographisches Puzzle. Dieses zu lösen kostet den Initiator einen gewissen Betrag an CPU-Zeit, der aufgebrachte Aufwand soll als Beweis für ein ernsthaftes Interesse an einer Verbindung betrachtet werden. Zudem ist *R1* von R signiert. Hierbei ist zu bemerken, dass zu diesem Zeitpunkt noch keine Zustandsänderung bei R eintritt, er muß sich den Beginn des Verbindungsaufbaus nicht merken! Nachdem I das ihm gestellte Puzzle gelöst hat, schickt er es zusammen mit der Antwort *I2* zurück (zusammen mit der zweiten Hälfte des Diffie-Hellman Schlüsseltauschs und dem eigenen public-Key) und signiert das Paket. Der Empfänger R kann nun ohne großen Aufwand die Lösung des Puzzles überprüfen. Ist sie richtig, so bestätigt er den Aufbau der Verbindung mit der *R2* Nachricht (wiederum signiert). Der Schlüsseltausch nach Diffie-Hellmandie läuft also parallel zur Authentifizierung. So entstehen *ESP (IPsec Encapsulated Security Payload)* Security Association in jede der beiden Richtungen.

4.2 Rendezvous Server

Um die HITs zu verwalten und eine Adresszuordnung bei sich gleichzeitig bewegenden Hosts zu gewährleisten, braucht eine HIP-Architektur eine Verwaltungsstelle, an der die aktuellen Zuordnungen von HIT zu Adresse gespeichert werden können. Diese so genannten *Redezvous Server* leiten HIP Pakete an die registrierten HIP Hosts weiter und ähneln in ihrer Funktion einem i3 Server.

Hierbei ist anzumerken, dass HIP nur für die Herstellung einer sicheren Verbindung genutzt wird. Sobald sich beide Hosts gegenseitig verifiziert haben, läuft die Kommunikation über IPsec, es handelt sich also tatsächlich um eine Ende-zu-Ende Verbindung. Einen Spiezialfall stellt die Kommunikation zwischen zwei mobilen Partnern dar. Hierbei muß der Rendezvous Server dauerhaft genutzt werden, da sich die Adressen beider Hosts gleichzeitig ändern können und sie deshalb einen gemeinsamen, beiden bekannten Punkt brauchen, an dem ihre HITs ihren aktuellen IP-Adressen zugeordnet werden können.

Ein HIP Host ist gegen DoS Angriffe jedoch nur sehr rudimentär geschützt, indem von einem Kommunikationsinitiator die Lösung eines kryptographischen Puzzles verlangt wird. Er hat nicht die Möglichkeit, sich aktiv gegen einen Angriff zu wehren, wie dies bei i3 der Fall ist. Mit anderen Worten: einem Angriff, der darauf abzielt den Host allein durch die große Datenmenge der vielen ankommenden Pakete zu lähmen wird hier weiterhin Erfolg haben, weil er sich direkt gegen den Host wenden kann.

5 Hi3

Bei oberflächlicher Betrachtung scheinen sowohl secure-i3 als auch HIP gut durchdachte Protokolle zu sein. Jedoch weisen beide gewisse Schwachstellen auf. Dies ist bei HIP unter anderem die (im Vergleich zu secure-i3) Anfälligkeit gegenüber DoS Angriffen und das Fehlen einer Möglichkeit solchen Angriffen aktiv zu begegnen.

Auf der anderen Seite steht bei secure-i3 nicht unbedingt die Sicherheit sondern vor allem die Robustheit im Vordergrund. Eine kryptographische Identifizierung und Verifizierung einzelner Host ist ebensowenig vorgesehen wie das senden verschlüsselter Nachrichten (auch

wenn Letzteres relativ einfach eingeführt werden könnte). Eine weitere „Schwachstelle“ der i3 basierenden Architekturen sind die relativ komplexen (verglichen mit IP) Router/i3-Server. Abgesehen davon, dass sie zunächst installiert werden müssen, ergeben sich eine Reihe von Sicherheitsbedenken. Ein Nutzer muß sich auf die Architektur verlassen und hat keine Möglichkeiten, aktiv für seine eigene Sicherheit zu sorgen. Selbst wenn man den guten Willen aller Beteiligten vorraussetzt (was man in einem offenen Netz wie dem Internet eigentlich nicht tun sollte) besteht aufgrund der Komplexität eines secure-i3-Servers eine gute Chance, dass sich bei der Implementierung Sicherheitslücken ergeben.

Da sich die Schwachstellen von secure-i3 und HIP erfreulich gut mit den Stärken des jeweils Anderen ausbessern lassen, scheint der Versuch angebracht, beide Protokolle zu einem zu fusionieren und dabei die Vorteile beider zu nutzen und die jeweiligen Schwachstellen abzufangen. Einen solchen Versuch stellt *Hi3* dar, er wird unter anderem in [GuJo04], [NiAO04] und [NiYW03] beschrieben.

5.1 Zusammenführung von i3 und HIP

Die Grundidee, die hinter der Zusammenführung von i3 und HIP zu Hi3 steht, ist eine i3 Architektur zu nutzen, um einen robusten HIP-Verbindungsaufbau zu ermöglichen. Dabei laufen alle HIP-Kontrollpakete über ein i3-Netz. Sobald beide Hosts sich gegenseitig authentifiziert haben, läuft die Datenverbindung wie von HIP gewohnt über IPsec.

Dabei übernimmt das i3 Netzwerk im Prinzip die Funktion des Rendezvous-Servers aus HIP. Je nach Implementierung könnten als Trigger-IDs sogar direkt die Hashwerte über die öffentlichen Schlüssel (HITs) benutzt werden, private Trigger werden dann nicht benötigt. Das Einfügen und Löschen von Triggern kann direkt über public-key Kryptographie abgesichert werden, der Schlüssel aus den Trigger-Constraints von secure-i3 kann entfallen.

5.2 Vorteile

Mit Hi3 erhält man eine Kombination der Vorteile aus i3 und HIP. Die gute Effizienz von HIP bleibt erhalten, da auch Hi3 letzten Endes eine Ende-zu-Ende Verbindung aufbaut. Dieser Aufbau erfolgt dabei gesichert (wie aus HIP gewohnt) und schützt gleichzeitig die Teilnehmer relativ gut vor DoS Angriffen (wie aus i3 bekannt), da die echten Adressen den Teilnehmern erst nach der Authentifizierung bekannt werden.

5.3 Verbindungsaufbau

Der Aufbau einer Verbindung erfolgt bei Hi3 in 8 Schritten, die in Bild 8 dargestellt sind. *Man beachte, dass die Implementierung in diesem Beispiel die von secure-i3 bekannten öffentlichen und privaten Trigger verwendet, die öffentlichen Trigger werden hierbei direkt aus den HITs generiert, die privaten Trigger sind der Allgemeinheit unbekannt.*

Zunächst stellt der Client seine HIP Init Anfrage (I1) an das i3-Netz. Dabei dient das HIT der Server als Adresse. Diese Nachricht wird von einem zufälligen i3-Node aufgefangen (hier S2) und an den Server weitergeleitet, der die öffentlichen Trigger des zu erreichenden Servers speichert (hier S1), die Zuordnung erfolgt dabei mit Hilfe von *Chord*, der Client wird über die Adresse von S1 unterrichtet um in Zukunft direkt den richtigen Server ansprechen zu können. Von hier wird die Nachricht an S3 weitergeleitet, weil hier der private Trigger des zu erreichenden Servers steht. Dieser leitet die Anfrage schließlich an den Server S weiter. Die Antwort (R1) erfolgt auf dem gleichen Weg zurück. Die zweite Stufe der Init Anfrage (I2) stellt der Client nun direkt an S1 und informiert diesen gleichzeitig über seinen eigenen

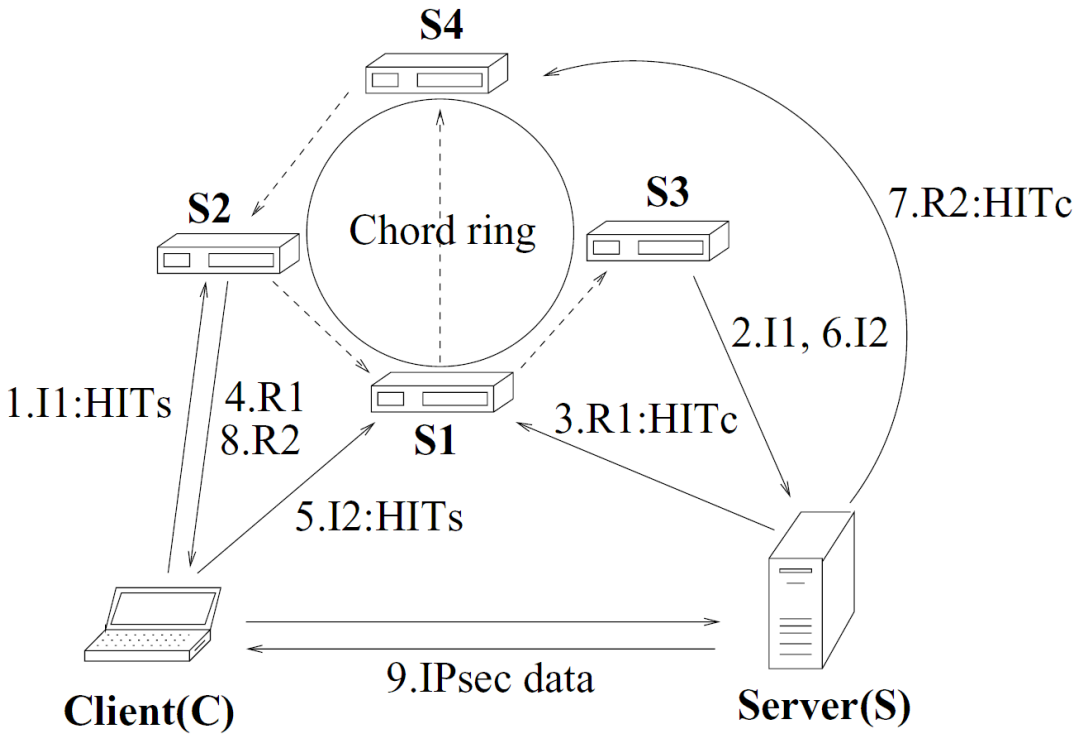

Abbildung 8: Verbindungsaufbau bei Hi3

öffentlichen Trigger, der sich auf S4 befindet. Die Nachricht wird über den privaten Trigger des Servers (S3) diesem zugestellt, seine Antwort (R2) geht nun wiederum direkt an den öffentlichen Trigger des Client C auf S4. Von hier wird die Nachricht an S2 weitergeleitet (hier liegt ja der private Trigger des Client). Sobald nun der HIP-Verbindungsaufbau (*HIP base exchange*, siehe auch 4.1) abgeschlossen ist, erfolgt der Datenaustausch IPsec-gesichert direkt zwischen Client und Server (wie von HIP gewohnt).

Da bei HIP eine R1-Nachricht noch keine Zustandsänderung beim Host erzeugt, wird in [NiAO04] vorgeschlagen die R1-Nachricht direkt von der Netzwerk-Infrastruktur erzeugen zu lassen. Dazu müsste der Server bei der Registrierung seines öffentlichen Triggers gleich ein paar vorberechnete R1 Nachrichten zur Verfügung stellen. Mit diesen R1-Nachrichten könnten auch neue private Trigger verknüpft werden die über das Netz verteilt werden. So kann einem Angriff, der darauf abziehlt, den Server mit I1-Nachrichten zu überfluten, besser begegnet werden. Weitergehend kann das Puzzle, das beim HIP-Verbindungsaufbau gestellt wird mit diesem privaten Trigger verknüpft werden, so dass die Antwort nur für diesen gültig ist. So würde erst eine erfolgreiche I2-Nachricht an den Host weitergeleitet werden. Dies entspricht prinzipiell dem in secure-i3 vorgeschlagenem DoS-Filter-Server, nur dass dieser hier auf das ganze Netz verteilt wird.

5.4 Mobilität

Die Komminikation von mobilen Hosts stellt immer besondere Anforderungen. Das Hauptproblem ist die sich verändernde Adresse eines mobilen Hosts. Solange die Kommunikation von einem mobilen Client und einem stationären Server betrachtet wird, ist das Problem noch relativ einfach zu lösen, der Client kann den Server bei jedem Wechsel über seine neue Adresse informieren. Dies ist möglich, weil der Server ja eine feste Adresse hat.

Anders sieht es aus, wenn es sich um zwei mobile Hosts handelt. Hier kann es passieren, dass beide gleizeitig ihre Adresse ändern. In einem solchen Fall können sie sich nicht gegenseitig über ihre neuen Adressen informieren, die Verbindung bricht ab. Die Lösung dieses Problems stellt ein Vermittler dar (der redezvous-server bei HIP, bei i3 jeder i3-Node), der eine feste Adresse hat und an den sich beide Hosts jederzeit wenden können.

Damit die Mobilität beider Kommunikationspartner gewährleistet bleibt, müssen die mobilen Hosts ihre Trigger bei jedem Adresswechsel mit ihrem neuen Aufenthaltsort aktualisieren. Um unnötigen Nachrichten-Overhead zu vermeiden, wird in [NiAO04] vorgeschlagen, dass die Hosts nur ihre öffentlichen Trigger selbst aktualisieren, alle privaten Trigger sind der Infrastruktur bekannt und können von dieser aktualisiert werden. Man beachte, dass diese Aktualisierung nur dann notwendig wird, wenn ein Host seine Adresse auch tatsächlich ändert.

Auf diese Weise wird die effiziente Datenübermittlung aus HIP (keine Dreiecksverbindungen für Daten) mit der Robustheit gepaart, die Hi3 von i3 geerbt hat.

6 Schlussbemerkung

Das Vorangeführte zeigt, dass eine Integration von i3 und HIP sinnvoll ist. Die sichere und effiziente Ende-zu-Ende Verbindung wird aus HIP übernommen. Das für die Kontroll-Eben verwendete secure-i3 stellt einen DoS Schutz ebenso zur Verfügung wie das *initial rendezvous* und die gleichzeitige Mobilität beider Endhosts.

Weiterhin zu beobachten bleiben die verschiedenen Möglichkeiten der Zuordnung von HITs zu IP-Adressen. In [NiAO04] wurden Versuche mit *Distributet Hash Tables (DHT)* gemacht und für gut befunden. Allerdings verliert Hi3 dadurch teilweise seine Robustheit gegenüber DoS Angriffen.

Abschließend ist noch einmal zu bemerken, dass Hi3 noch kein fertiges Protokoll ist, sondern lediglich die Idee der Verknüpfung zweier existierender Protokolle. Dabei können und werden verschiedene Wege beschritten, eine finale und einheitliche Lösung ist jedoch noch nicht bekannt.

Literatur

[ALPS03] Daniel Adkins, Karthik Lakshminarayanan, Adrian Perrig und Ion Stoica. Towards a More Functional and Secure Network Infrastructure. Technischer Bericht UCB/CSD-03-1242, EECS Department, University of California, Berkeley, 2003.

[GuJo04] Andrei Gurtov und Anthony D. Joseph. *Friends or Rivals: Insights from Integrating HIP and i3.* Helsinki Institute for Information Technology, Helsinki, Finnland. 2004.

[Mänt] Martti Mäntylä? How HIP Works.

[NiAO04] Pekka Nikander, Jari Arkko und Börje Ohlman. *Host Identity Indirection Infrastructure (Hi3).* Ericsson Research Nomadiclab/IP Networks, Helsinki, Finnland / Stockholm, Schweden. 2004.

[NieH05] P. Nikander, P. Jokela (editor) und T. Henderson. Host Identity Protocol draft-ietf-hip-base-04. Network Working Group Internet Draft 04, ICSAlabs, a Division of TruSecure Corporation and Ericsson Research NomadicLab and The Boeing Company, Oktober 2005.

[NiYW03] Pekka Nikander, Jukka Ylitalo und Jorma Wall. Integrating Security, Mobility and Multi-Homing in a HIP Way. In *NDSS*, 2003.

[SAZS+02] I. Stoica, D. Adkins, S. Zhuang, S. Shenker und S. Surana. Internet Indirection Infrastructure, 2002.

Abbildungsverzeichnis