Technische Universität Ilmenau

Fakultät für Informatik und Automatisierung

Diplomarbeit

Implementierung und Bewertung von L-MIFA im Netzwerksimulator NS-2

Inventarisierungsnummer: 2007-04-16/054/II00/2235

---

vorgelegt von:	Christian Kellner
eingereicht am:	16. 07. 2007
geboren am:	16. 01. 1975 in Jena
Studiengang:	Ingenieurinformatik
Studienrichtung:	Multimediale Informations- und Kommunikationssysteme
Anfertigung im Fachgebiet:	Integrierte Hard- und Softwaresysteme Fakultät für Informatik und Automatisierung
Verantwortlicher Professor:	Prof. Dr. Ing. habil. Andreas Mitschele-Thiel
Wissenschaftlicher Betreuer:	Dipl.-Ing. Ali Diab

Danksagung

... Obwohl ich natürlich alleiniger Verfasser dieser Arbeit und allein dafür verantwortlich bin, möchte ich mich an dieser Stelle herzlich bei meinem Betreuer Ali Diab für die unermüdliche Unterstützung beim Verfassen der Arbeit bedanken. Des weiteren danke ich mich meiner Familie nicht nur für die finanzielle, sondern auch moralische Unterstützung. ...

Kurzfassung

. . . All-IP Netze finden zunehmend wachsende Verbreitung. Diese vielfältigen Kommunikationsnetze sind durch ein gemeinsames IP Core verbunden. Dies ermöglicht dem Nutzer unabhängig vom Aufenthaltsort online zu bleiben. Diese Netzwerke erfreuen sich wachsender Beliebtheit. Jedoch sind eine Vielzahl von Problemen bis heute ungelöst. Eine der Hauptaufgaben ist ein schneller und nahtloser Übergang zwischen den Zugangspunkten des Netzwerkes. Low Latency Mobile IP Fast Authentication Protocol (L-MIFA) vermeidet Probleme von Mobile IP (MIP). L-MIFA erfüllt Echtzeitanforderungen ohne Verwendung spezialisierter Router oder anderer Einschränkungen der Netzwerktopologie. In dieser Arbeit wird die Performanz von L-MIFA bewertet und mit MIP, Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) und Mobile IP Fast Authentication (MIFA) verglichen. Die Bewertung erfolgt im Netzwerksimulator 2 (ns-2). Alle vier Protokolle werden in derselben Topologie betrachtet.

Die Simulationsergebnisse haben gezeigt, dass bei L-MIFA und MIFA die Handoff Latenzzeiten und Paketverlustraten in Up- und Downlink deutlich geringer sind als bei HAWAII und MIP. Die wird dadurch erreicht, dass L-MIFA und MIFA sich nur mit dem neuen Foreign Agent (FA) zu registrieren brauchen, um wieder Pakete senden zu können. MIP hingegen muss sich bei dem Home Agent (HA) registrieren. HAWAII muss die neue Postion dem alten FA mitteilen. Für Downlink-Quellen erzielen HAWAII und MIFA vergleichbare Werte. Die Handoff Latenzzeiten und Paketverlustraten von L-MIFA sind nahe 0 und somit besser als MIFA in Up- und Downlink. . . .

Abstract

. . . All-IP networks become increasingly visible. The various communication networks are aimed to be connected with each other through a common IP core, so that the user will stay always online, anytime and anywhere. We believe that these networks will be the popular network in the future. However a lot of challenges remain unsolved until today. One of the major challenges is how to achieve a seamless and fast handoff while moving from one point of attachment to another. Low Latency Mobile IP Fast Authentication Protocol (L-MIFA) is proposed to avoid the problems of Mobile IP (MIP) and to match the real-time requirements without introducing intermediate nodes and without making any restriction on the network topology. In this work we evaluate the performance of L-MIFA compared to MIP, Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) and Mobile IP Fast Authentication (MIFA) . The evaluation is performed by means of network simulator 2 (ns-2). The four protocols are evaluated deploying the same topology.

Our simulation results have shown that L-MIFA and MIFA outperforms HAWAII and MIP with respect to handoff latency and the number of dropped packets for uplink and downlink traffic. This is because L-MIFA and MIFA needs only to register with the new Foreign Agent (FA) to be able to resume sending of packets, while MIP needs to register with the Home Agent (HA) and HAWAII requires updating the new location at the old FA. For downlink traffic HAWAII and MIFA perform comparable to each other. The handoff latency and packet dropping of L-MIFA is near zero and better than MIFA in up- and downlink. . . .

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

All-IP-Netzwerke finden wachsende Verbreitung. So sind in den letzten Jahren WLAN-Hotspots in großer Zahl entstanden. Bis zum Jahr 2009 werden für Europa 35000 aktive WLAN-Hotspots [2] prognostiziert. Auch UMTS findet, wenn auch nur schleppend, Verbreitung. Damit geht die Bedeutung drahtgebundener Internetzugänge zurück. Nicht nur Laptops ermöglichen den mobilen Onlinezugang, sondern auch kleine Geräte wie z. B. VoIP-Telefone [1]. Diese stellen eine ernst zunehmende Konkurrenz für Mobilfunkprovider dar, da sie wesentlich kostengünstiger sind. Es ist zu erwarten, dass All-IP-Netze in Zukunft die am verbreitetesten Netzwerke sein werden.
Mittels All-IP-Netzen stehen Dienste wie VoIP, IPTV (Internetfernsehen), Onlinespiele, multimediale Emails oder Viedeostreaming dem Nutzer zu jeder Zeit an jedem beliebigen Ort zur Verfügung. Eine Hauptaufgabe des All-IP-Netzes ist die Verwaltung der Mobilität des Teilnehmers. Dazu werden spezielle Mobilitätsprotokolle benötigt. Um den Benutzer auch bei Veränderung Aufenthaltsortes den ständigen Zugang zu gewährleisten, ist es notwendig, den Zugangspunkt zum Netz schnell wechseln zu können. Bisherige Mobilitätsprotokolle wie Mobile IP reichen dazu nicht aus, da sie keine Echtzeitanforderungen erfüllen. Somit ist die Notwendigkeit zur Entwicklung neuer Protokolle gegeben.

1.2 Aufgabenstellung

Gegenstand dieser Arbeit ist die Implementierung und Bewertung des Mobilitätsprotokolls L-MIFA im Netzwerksimulator NS-2. L-MIFA steht für „Low Latency Mobile IP Fast Authentification Protocol“ und ist eine Weiterentwicklung von „Mobile IP Fast Authentication Protocol“ (MIFA), das am Fachgebiet „Integrierte Hard- und Softwaresysteme“ der Technischen Universität Ilmenau entwickelt wurde.
L-MIFA ist in ns-2 zu implementieren. Anschließend soll das Protokoll bewertet und mit anderen Protokollen verglichen werden. Hierzu werden die Handoff Latenzzeiten

und Paketverlustraten gemessen. Anschließend werden sowohl die Durchschnittswerte als auch die statistischen Verteilungsfunktionen der Handoff Latenzzeiten und Paketverlustraten grafisch dargestellt. Zum Abschluß der Arbeit ist zu bewerten, inwiefern die Verkürzung der Handoff Latenzzeit und die Minimierung der Paketverlustraten durch das Protokoll L-MIFA erreicht wurde. Weiterhin soll geklärt werden, welches der untersuchten Protokolle Echtzeitforderungen erfüllt. Für Echtzeitanwendungen muss die Handoff Latenzzeit je nach Anwendung zwischen 50 und 100 ms liegen.

1.3 Aufbau der Arbeit

Die folgende Arbeit ist folgendermaßen gegliedert. Kapitel 2 beschreibt die Mobilitätsprotokolle und beleuchtet Stand der Technik. Am Ende des Kapitels werden die Vorteile und Nachteile dieser Protokolle in tabellarischer Form aufgelistet. Kapitel 3 liefert eine detaillierte Beschreibung von MIFA und L-MIFA. NS-2 ist kurz beschrieben im Kapitel 4. Kapitel 5 erläutert die Implementierung von L-MIFA im NS-2 und beschreibt die benutzten Szenarien für die Bewertung und den Vergleich mit anderen Mobilitätsprotokollen. Kapitel 6 analysiert die Ergebnisse. Eine Zusammenfassung und einen Ausblick ist gegeben im Kapitel 7.
Am Ende der Arbeit befinden sich die Anhänge. In Anhang A befinden sich die Szenarioeinstellungen für HAWAII, MIP, MIFA und L-MIFA. Anhang B enthält die wichtigstens Prozeduren für L-MIFA.

2 Stand der Technik und Mobilitätsprotokolle

In diesem Kapitel soll ein Überblick über den Stand der Technik verschafft werden. Zuerst werden grundlegende Begriffe des Mobilitätsmanagement erklärt. Kriterien für den Vergleich von Mobilitätsprotokollen werden aufgestellt und anschließend einzelne Mobilitätsprotokolle in ihrer Funktionsweise vorgestellt. Dabei sollen Vor- und Nachteile dieser Protokolle, die in einer Tabelle aufeglistet werden, herausgefunden und Verbesserungspotentiale aufgezeigt werden.

2.1 Grundlagen

Zunächst sollen einige Grundbegriffe geklärt werden, die für das Verständnis von Handoff-Vorgängen und des Mobilitätsmanagements unbedingt erforderlich sind.

Definition Handoff Bei einem Wechsel des Access-Point der Mobile bei aktiver Datenübertragung spricht man von einem *Layer-2 Handoff* (L2). Wird dabei über den Foreign Agent in eine anderes Subnetz gewechselt, handelt es sich dabei um einen *Layer-3 Handoff* (L3). In dieser Arbeit werden grundsätzlich nur L3 Handoffs untersucht, da im Netzwerksimulator ns-2 keine L2 Handoffs implementiert sind.

Hard Handoff Bei einem vollständigen Wechsel des Access Points der Mobile spricht von einem Hard Handoff. Bei dieser Art des Handoffs kann die Mobile jeweils nur von einer Basisstation Pakete empfangen bzw. an diese senden.

Soft Handoff Bei einem Soft Handoff empfängt bzw. sendet die Mobile Pakete von beiden Basisstationen. Bei dieser Art des Handoffs sind die Paketverluste geringer als bei einem Hard Handoff.

Handoff Latenzzeit Unter der Handoff Latenzzeit versteht man die Zeitdauer, in der die Mobile, ausgelöst durch einen Handoff, keine Pakete senden oder empfangen kann. Diese Zeitdauer ist je nach verwendeten Protokoll unterschiedlich definiert.

Movement Detection Unter der *Movement Detection* Zeit [19, S. 2] versteht man die Zeitdauer zwischen dem Verbindungsabbruch mit der alten Basisstation und dem Empfang des Advertisement Pakets von der neuen Basisstation. Diese Zeitdauer kann nicht gemessen werden. Man unterscheidet zwei grundlegende Verfahren:

- Eager Cell Switching (ECS)
- Lazy Cell switching (LCS)

Bei ECS meldet sich die Mobile sofort nach Empfang des neuen Advertisement Pakets um zur neuen Basisstation, wobei davon ausgegegangen wird, dass sich die Mobile in eine bestimmte Richtung bewegt. Dadurch werden kurze Latenzzeiten erzielt. Nachteil diese Verfahrens ist, dass bei langsamer Bewegung der Mobile Ping-Pong-Effekte auftreten, d. h. die Mobile meldet sich mehrfach bei alter und neuer Basisstation an.
Bei LCS wartet die Mobile den Empfang mehrerer Advertisement Pakte ab, ehe diese sich ummeldet. Dadurch vergrößert sich einerseits die Handoff Latenzzeit, aber andererseits werden Ping-Pong-Effekte vermieden.

2.2 Mobilitätsprotokolle

Mobilitätsprotokolle übernehmen das Management von mobilen Endgeräten. Sie verwalten die Kommunikation aller beteiligten Komponenten. Dieser Vorgang wird auch Signalisierung genannt.
Aufgrund ihrer Granularität werden Mobilitätsprotokolle in Mikro- und Makromobilitätsprotokolle unterschieden. Die *Makromobilitätsprotokolle* verwalten die Mobilität zwischen Heimatnetz und dem mobilen Endgerät. Die Domäne des Netzes kann bei einem Handoff komplett gewechselt werden (eng. inter domain). Die Verwaltung der Kommunikation mit dem entfernten Heimatnetz ist aufwendig, weshalb diese Protokolle nicht für einen schnellen Standortwechsel und der damit verbundenen Handoff geeignet sind (Abbildung 2.1).
Aus diesem Grund wurden *Mikromobilitätsprotokolle* entwickelt, die die lokale Mobilität des Teilnehmers besser verwalten. Diese Protokolle unterstützen Handoff nur

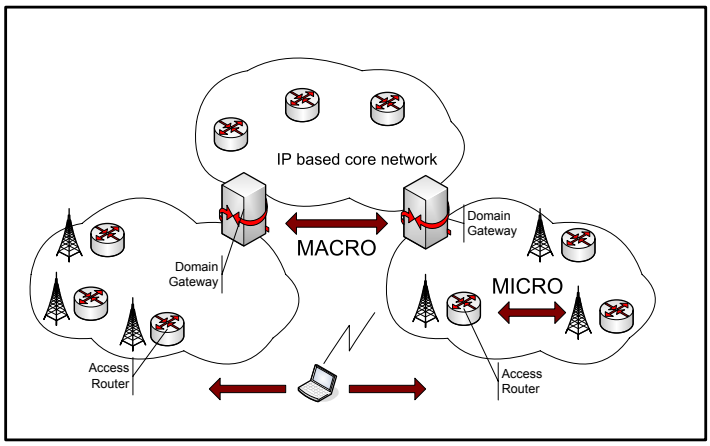

Abbildung 2.1: Struktur eines all-IP basierten Netzwerks

innerhalb einer Domäne (eng. intra domain). Sie werden unter folgenden Aspekten betrachtet:

- Hierarchische Mobilität: Anordnung des Netzes, die den Verkehr zum Home Agent entlastet, Beispiele dafür sind Hierarchical Mobile IP, Celluar IP und HAWAII
- Hierarchisches Tunneln: Ein verteiltes Netz von Foreign Agents verwaltet den Paketverkehr, ein Beispiel dafür ist Hierarchical Mobile IP
- Mobile-Specific Routing: Lokal werden spezielle Routing-Techniken verwendet, Beispiele dafür sind Celluar IP und HAWAII

In den folgenden Abschnitten werden Protokolle in ihrer Funktionalität vorgestellt und Mobilitätsaspekte betrachtet.

2.3 Kriterien für den Vergleich von Mikromobilitätsprotokollen

Mikromobilitätsprotokolle sollen einen möglichst unterbrechungsfreien Datenstrom gewährleisten und Netzwerkressourcen effizient nutzen. Zudem sollen möglichst viele mobile Geräte innerhalb einer Domäne bedient werden können. Folgende Eigenschaften sind wichtige Gütekriterien [8] [12]:

Handoff Management ist der wichtigste Aspekt des Mobility Management. Hauptsächlich betrachtet werden Layer-3 Handoff-Vorgänge:

- Handoff Management Parameter: Interaktionen mit der physikalischen Schicht, Initiierung von Handoff-Mechanismen, usw.
- Handoff Latenzzeit: die benötigte Zeit zur Vervollständigung des Handoff innerhalb einer Domäne.
- Potentielle Paketverluste: verlorene Pakete bei Handoff
- Beteiligte Stationen: Anzahl der Mobilitäts-Agenten (MA), welche die Routing Tabelle und Prozessnachrichten aktualisieren

Passive Konnektivität und Paging Bei aktiver Konnektivität wird ständig der Aufentshaltort der Mobile ermittelt. Dadurch entsteht sehr viel Netzwerkverkehr, der für die Signalisierung benötigt wird. Um diesen Verkehr zu senken und Energie der Mobile zu sparen, wird *passive Konnektivität* verwendet. Diese definiert in drahtlosen Netzen energiesparende Betriebsmodi für Mobile, die gerade keine Daten senden oder empfangen. In den meisten Fällen bedeutet das in der Praxis die Konstruktion einer *Paging Architektur*, die das Netzwerk in geografische Zonen, so genannte *Paging Areas*, einteilt.

Intra Network Traffic Dieses Verfahren beschreibt den Paketaustausch zwischen Mobilen innerhalb der selben Domäne. Nur zu einander korrespondierte Mobile können Daten austauschen.

Skalierbarkeit und Robustheit Eine wichtige Anforderung drahtloser Netzwerke ist die sichere Kommunikation möglichst vieler Mobile. Zudem soll das Netzwerk robust gegenüber Störungen sein.

Sicherheit Ein wichtiges Ziel für Mobilitätsprotokolle ist die Implementierung eines Sicherheitsmodells für Authentication, Authorization and Account (AAA).

QoS Mikromobilitätsprotokolle sollen die Unterstützung verschiedender QoS-Klassen erlauben, angefangen von best effort bis hin zu Echtzeitanwendungen.

2.4 Mobilitätsprotokolle

In diesem Abschnitt werden die Mobilitätsprotokolle in ihrer Funktionsweise vorgestellt und verglichen. Vor- und Nachteile dieser Mobilitätsprotokolle sollen herausgestrichen und tabellarisch aufgelistet werden, um so Verbesserungspotentiale aufzuzeigen.

2.5 Mobile IP

Mobile IP (MIP) [6] ist ein Makromobilitätsprotokoll. Ein mobiles Endgerät wird in diesem Zusammenhang als *Mobile Node* (MN) bezeichnet und gehört zu einem Heimatnetz, indem der *Home Agent* (HA) als Ansprechpartner fungiert. Daten werden zwischen dem Endgerät und dem Kommunikationspartner, dem *Correspondent Node* (CN), ausgetauscht. Befindet sich die Mobile nicht im Heimatnetz, so können die Daten nicht direkt zugestellt werden. Der Mobile Node erhält von der Mobilitätsverwaltungskomponente des Fremdnetzes, dem *Foreign Agent* (FA), eine neue *Care-of Address* (CoA). Die Daten werden vom CN zum MN geschickt. Der HA prüft die aktuelle Adresse des MN. Wenn der MN sich nicht im Heimatnetz befindet, tunnelt der HA die Datenpakete zum MN über den aktuellen FA. Auf dieser Art und Weise ist der MN transparent zum CN.

2.5.1 Registrierung

Vorraussetzung für das Senden und Empfangen von Daten ist die Registrierung der Mobile in dem aktuellen Netz, in dem die Mobile einen funktechnischen Zugang hat. Als erstes muß die Mobile [6, S. 28] vom aktuellen Home- oder Foreign Agent eine *Advertisement Message* erhalten, welche zyklisch ausgesendet werden. Sowie der MN diese erhalten hat, sendet er einen *Registration Request* an den aktuellen Agenten. Dieser wird an den Home Agent weitergeleitet und mit einem *Registration Reply* beantwortet. Sobald die Mobile den Reply erhalten hat, können Daten gesendet und empfangen werden. Abbildung 2.2 zeigt den genauen Ablauf der Registrierungsprozedur am Beispiel einer Downlinkverbindung zwischen dem Correspondent Node und dem Mobile Node. Diese Registrierungs-Prozedur findet bei Erstanmeldung der Mobile in einem Netz und bei jedem Handoff statt.

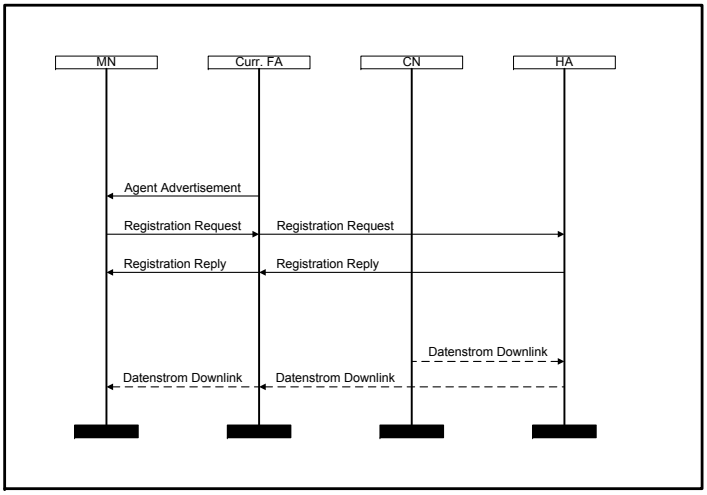

Abbildung 2.2: Registrierung in Mobile IP - Downlink

2.6 Low Latency Handoff in Mobile IPv4

Mobile Ipv4 beschreibt Handoff-Techniken [16] im Ipv4-Layer. Handoffs erfolgen zwischen verschiedenen Subnetzen, die durch unterschiedliche Foreign Agents versorgt werden. In bestimmten Fällen liegt die Handoff Latenzzeit über einem Schwellwert, der für zeitkritische Anwendungen eingehalten werden muss. Mittels zweier Verfahrensansätze, der *Pre-Registration* und der *Post-Registration*, kann die Handoff Latenzzeit verkürzt werden. Mittels Pre-Registration ist die Kommunikation der Mobile mit dem neuen Foreign Agent, wenn dieser noch mit dem alten FA (oFA) verbunden ist. Mit Hilfe der Post-Registration erfolgt die Bereitstellung der Daten am neuen FA (nFA), bevor die Registrierung der Mobile am nFA abgeschlossen ist. Beide Methoden können kombiniert werden, um den Handoff-Vorgang noch weiter zu verbessern.
Der ursprüngliche Ansatz von Mobile IP betrachtet keine bestimmten Schichten des Protokoll-Stacks, so dass klar zwischen Layer-2 (L2) und Layer-3 (L3) unterschieden werden muss. Dies wirkt sich nachteilig auf die Handoff Latenzzeit aus. Eine Registrierung beim nFA kann erst erfolgen, nachdem der L2 Handoff abgeschlossen ist. Während der Registrierungsphase können keine Pakete gesendet oder empfangen werden.

2.7 Ausblick - Mobile IPv6

Mobile IP in der Version 6 [11] ist eine Mobilitätserweiterung des wenig verbreiteten IPv6-Standards und bietet verbesserte Mobilitätsansätze gegenüber MIPv4. Es werden

keine Foreign Agents benötigt. Routenoptimierung verhindert Triangle Routing, die auch ohne Secure Association arbeitet. MIPv6 verwendet auch Verbesserungen des Ipv6-Protokolls, die die Robustheit des Netzes erhöhen. Beispielsweise sichert der *IPv6 Neighbor Unreachability Detection* Algorithmus [11, S. 6] die Erreichbarkeit zwischen Mobilen und Router. Des weiteren reduziert sich der Protokolloverhead durch durch Verwendung des *IPv6 Routing Header* anstatt der IP-Kapselung.

2.7.1 MIPv6 Handoff Prozedur

Mobile IPv6 bietet transparente Host-Mobilität [18, S. 11] innerhalb IPv6.

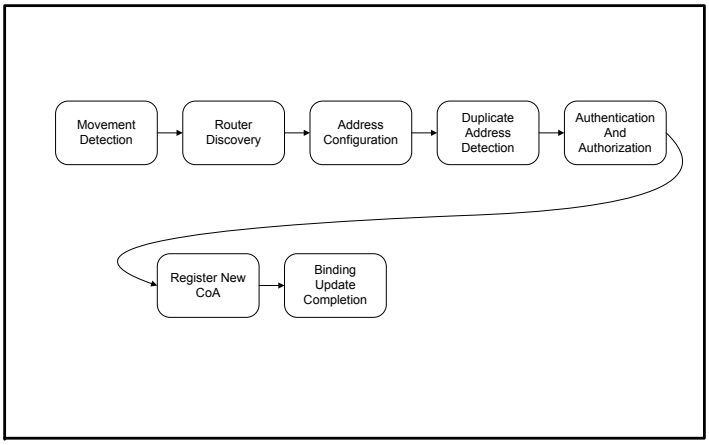

Abbildung 2.3: MIPv6 Handoff Prozedur

Das Protokoll ermöglicht der Mobile den Wechsel des Netzwerkes ohne Änderung der IPv6-Adresse. Die Mobile ist immer über die Heimatadresse (eng. home address) adressierbar, über die sie im Heimatnetz gemeldet ist. Befindet sich die Mobile außerhalb des Heimatnetzes, können Pakete mittels der Heimatadresse geroutet werden. Somit ist die Bewegung der Mobile unsichtbar in der Transportschicht und anderen höheren Protokollschichten.
Abbildung 2.3 zeigt den grundsätzlichen Ablauf der Handoff Prozedur. Diese Prozedur wird ausgeführt, wenn die Mobile den Zugangspunkt zum Internet in ein anderes IPv6-Netzwerk wechselt. Dieser Vorgang wird als *Roaming* bezeichnet. Bis auf die Prozedur „Authentication and Authorization“ sind alle Einzelprozeduren notwendig zur Durchführung des Handoff.

2.7.2 Vergleich zwischen MIPv4 und MIPv6

In MIPv6 [11, S. 6] werden keine spezialisierten Router wie z. B. Foreign Agents wie in MIPv4 benötigt. MIPv6 operiert in jeder Umgebung, ohne dass der Router MIPv6 speziell unterstützen muss. Die Routenoptimierung ist ein fundamentaler Bestandteil des Protokolls, mit dessen Hilfe ohne vorher gesetzte Secure Association gearbeitet werden kann. Für das Versenden von Paketen wird der IPv6 Routing Header verwendet anstatt der IP-Kapselung. MIPv6 ist von speziellen Link Layern entkoppelt, da es den den *IPv6 Neighbor Discovery* Mechanismus anstatt des ARP-Protokolls benutzt. Der *Dynamic Address Home Agent Discovery* Mechanismus ermöglicht das Ansprechen einer einzelnen Mobile, ohne wie bei IPv4 einen Broadcast verwenden zu müssen.

2.8 Hierarchical Mobile IP

Hierarchical Mobile IP [7] ist eine Mikromobilitätserweiterung von Mobile IP. Bei Mobile IP ändert sich bei jeder Registrierung mit dem Home Agent die Care-of Address. Bei zu großer Entfernung zwischen Besuchernetz und Heimatnetz wird die Verzögerungszeit der Registrierung zu groß, die für die Signalisierung zwischen Mobile und Home Agent benötigt wird.
Um dieses Problem zu lösen, wird eine lokale Registrierung eingeführt. Diese verringert die Anzahl der Signalisierungsnachrichten, die zum Heimatnetz gesendet werden müssen. Außerdem verkürzt sich die Zeitdauer des Registrierungsvorgangs für den Fall, das die Mobile den FA innerhalb des Besucher-Netzes wechselt.

2.8.1 Aufbau eines HMIP-Netzwerkes

Mobile IP wird hierarchisch erweitert. Auf der höchsten Ebene steht der *Gateway Foreign Agent* (GFA), der die Kommunikation innerhalb des Besuchernetzes verwaltet und Informationen mit dem Home Agent des Heimatnetzes austauscht. In der Ebene darunter befinden sich die *Regional Foreign Agents* (RFA). In der untersten Ebene befinden sich die lokalen Foreign Agents (FA), die die Kommunikation mit der Mobile realisieren. Findet nun ein Handoff statt, so verwaltet der nächste erreichbare RFA oder GFA diesen Vorgang, wodurch die Verzögerung der Signalisierung verringert werden kann. Abbildung 2.4 zeigt die typische Struktur eines HMIP-Netzwerkes.

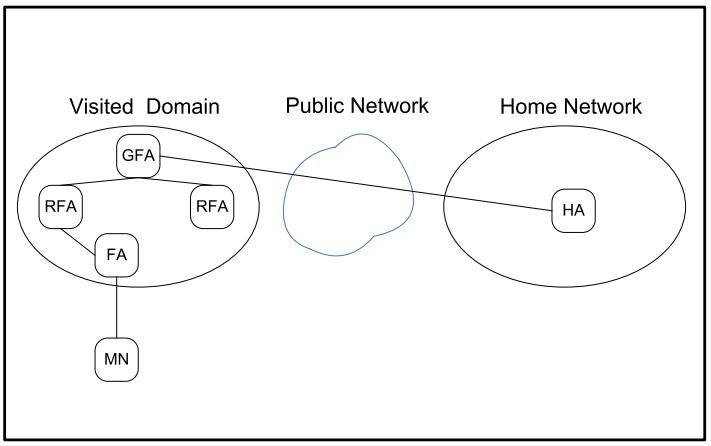

Abbildung 2.4: Struktur eines HMIP-Netzwerkes

2.8.2 Signalisierung

Sobald die Mobile das Besuchernetz erstmalig erreicht, wird eine Registrierung mit dem Home Agent des Heimatnetzes (eng. home registration) durchgeführt. Die Home Registration wird auch bei Wechsel des GFA durchgeführt. Der Home Agent erzeugt daraufhin einen *Registrierungsschlüssel* (eng. registration key), der an die Mobile und das Besuchernetz weitergeleitet wird. Dieser Schlüssel kann ab diesem Zeitpunkt für die Authentfizierung der lokalen Registrierung innerhalb des Besucher-Netzes verwendet werden.

Regional Registration Bei einer Home Registration speichert der Home Agent die Care-of Address (CoA) der Mobile, bei regionaler Registrierung die CoA des GFA. Der Gateway Foreign Agent speichert in einer Liste alle im Besuchernetz aktuell registrierten mobilen Endgeräte. Die im HA gespeicherte CoA ändert sich nicht, solange die Mobile den FA unterhalb desselben GFA wechselt, lediglich die lokale CoA des GFA wird geändert. In diesem Fall braucht der HA nicht informiert zu werden, wodurch sich die Anzahl der Signalisierungsnachrichten verringert und der Flaschenhals zum HA entlastet wird. Für die lokale Registrierung werden jeweils ein lokaler Regional Registration Request und Registration Reply verwendet. In Abbildung 2.5 [7, S. 5] ist der Signalfluss bei der Home- und Regional Registration Prozedur zu sehen.

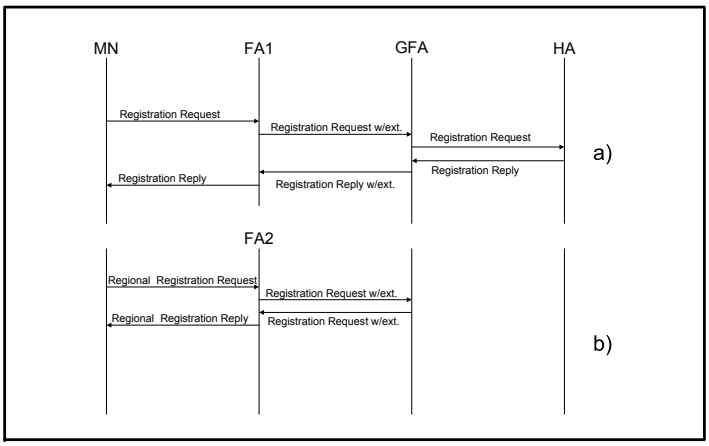

Abbildung 2.5: a) Home Registration, b) Regional Registration

2.8.3 Zusammenfassung

Mittels HMIP kann die Mikromobilität verbessert werden. Die Anzahl der Signalisierungsnachrichten verringert sich durch den Einsatz der lokalen Registrierung, wodurch auch Handoff Latenzzeiten verkürzt werden können. Die Realisierung erfordert eine Infrastruktur von GFAs und RFAs, wodurch sich der Verwaltungsaufwand für das Netzwerk erhöht.

2.9 Celluar IP

Das Celluar IP Protokoll [9] ist eine Mikromobilitätserweiterung von Mobile IP. Entwickelt wurde es von der Columbia University in New York und Ericsson Research. 1998/99 wurde das Protokoll bei der IETF eingereicht. Es unterstützt lokale Mobilität und Paging sowie Hard- und Semisoft-Soft Handoff. Durch die Verwendung eines Gateway Routers ist bei lokaler Mobilität keine erneute Registrierung beim Home Agent erforderlich. Alle an der Kommunikation beteiligten Knoten sammeln Weginformationen der Daten der Mobile führen. Um dieses Ziel zu erreichen, werden vier grundlegende Entwurfsprinzipien verfolgt:

1. Standortinformationen werden in verteilten Datenbanken gespeichert
2. Standortinformationen einer Mobile werden durch IP-Datagrams erzeugt und aktualisiert

3. Standortinformationen werden “Soft-State“ abgespeichert

4. Location Management für inaktive und aktive Mobile findet getrennt statt

Ein besonderer Augenmerk bei dem Entwurf von Celluar IP lag darauf, schnelle und ungewöhnliche Bewegungen von Mobilen gleichermaßen zu unterstützen wie statische Mobile.

2.9.1 Aufbau und Arbeitsweise

An oberster Stelle dieses Netzwerkes steht der Internet Backbone mit Internetzugriff, der Mobile IP unterstützt. Ihm angeschlossen sind die Gateways (GW), die den Zugriff auf die einzelnen Celluar IP Netzwerke ermöglichen. Die Mobile Hosts (MH) stellen die Verbindung über Basisstationen (BS) her, die jeweils einem Celluar IP Netzwerk angeschlossen sind. Die Celluar IP Knoten des Netzwerkes verfügen jeweils über einen Uplink- und Downlink Neighbor. Diese Struktur ist vorkonfiguriert wird durch den „Neighbor Selection Algorithm“ erstellt. Abbildung 2.6 [9, S. 5] zeigt schematisch den typischen Aufbau eine Celluar IP Netzwerkes.

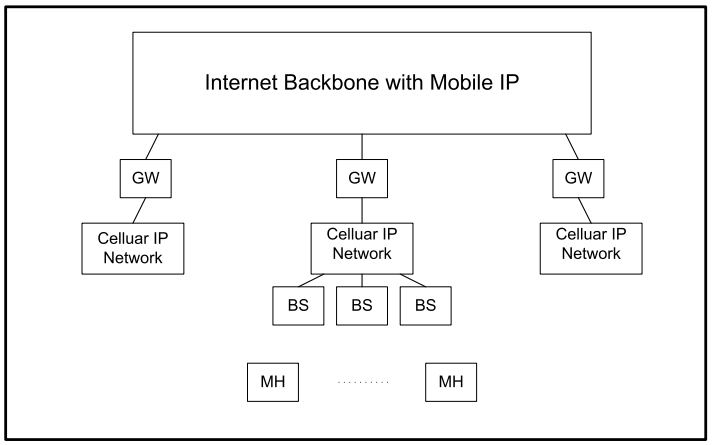

Abbildung 2.6: Struktur Celluar IP mit Internetzugriff

Die Verbindungsherstellung der Mobile läuft wie folgt ab: Die Basisstionen senden periodisch ein Beacon-Signal aus. Die Mobile nutzt dieses Beacon-Signal zur Standortermittlung der nächstgelegenen Basisstation, von welcher sie Pakete empfangen oder an

sie senden kann. Unabhängig von der Zieladresse werden die IP-Pakete „hop-by-hop“ von der Basisstation zum Gateway geroutet. Dabei wird der kürzeste Pfad gewählt. Das Routing von IP-Paketen, die an die Mobile adressiert sind, erfolgt über eine Kette von *chached mapping*. Zur Vermeidung von Time Outs verwendet Celluar IP ICMP-Steuerpakete. Diese werden periodisch ausgesendet und enthalten Authentifizierungsinformationen.

2.9.2 Lokalisierung und Routing

Celluar IP verwendet zwei parallele Cache-Systeme: den *Paging Cache* und den *Route Cache*. Diese beiden Systeme haben grundsätzlich die selbe Arbeitsweise.
Das Netzwerk muß in der Lage sein, den Weg einer aktiven Mobile von Basisstation zu Basisstation zu verfolgen, ohne die Mobile suchen zu müssen. Das beinhaltet auch auftretende Handoffs. Nur auf diese Weise ist das Zustellen von Paketen gesichert. Für inaktive Mobile wird die Kommunikation auf ein Minimum beschränkt, wodurch die Batterie des mobilen Endgerätes geschont wird. Die getrennte Lokalisierung von aktiven und inaktiven Mobilen bewirkt zusätzlich eine Performanzsteigerung und eine bessere Skalierbarkeit des Netzes.

Lokalisierungs Management Celluar IP ermöglicht die Bewegung von Mobilen innerhalb geographisch großer Flächen. Mehrere aneinander angrenzende Zellen werden zu *Paging Areas* zusammengefasst. Diese haben in jedem Celluar IP Netzwerk eine eindeutige Kennung. Jede Basisstation überträgt diesen *Paging Area Identifier* in dem periodisch ausgesendeten Beacon-Signal. Dadurch erkennt die Mobile das Betreten einer neuen Paging Area.
Falls eine inaktive Mobile eine neue Paging Area betritt, muß sie ein *Paging Update Paket* an die aktuelle Basistation senden, welches „hop-by-hop“ an den zugehörigen Gateway geroutet wird. Falls diese Mobile sich nur innerhalb der Paging Area bewegt, wird nur nach Ablauf der Paging Update Time ein Paging Update Paket gesendet. Einige ausgewählte Knoten des Celluar IP Netzwerkes sind mit einem Paging Cache ausgestattet, welche Paging Update Pakete überwachen und an die neue Paging Area weiterleiten. Diese Pakete verlieren ihre Gültigkeit bei Erreichen des Gateways.
Sobald eine Mobile Paging Paket erhält, wird sie aktiv und erstellt ein den Route Cache und sendet ein Route Update Paket. Nachfolgende IP-Pakete, die an den selben Host adressiert sind, werden über diesen Route Cache geroutet. Abbildung2.7 [9, S. 6] zeigt den Vorgang der Lokaliesierung.

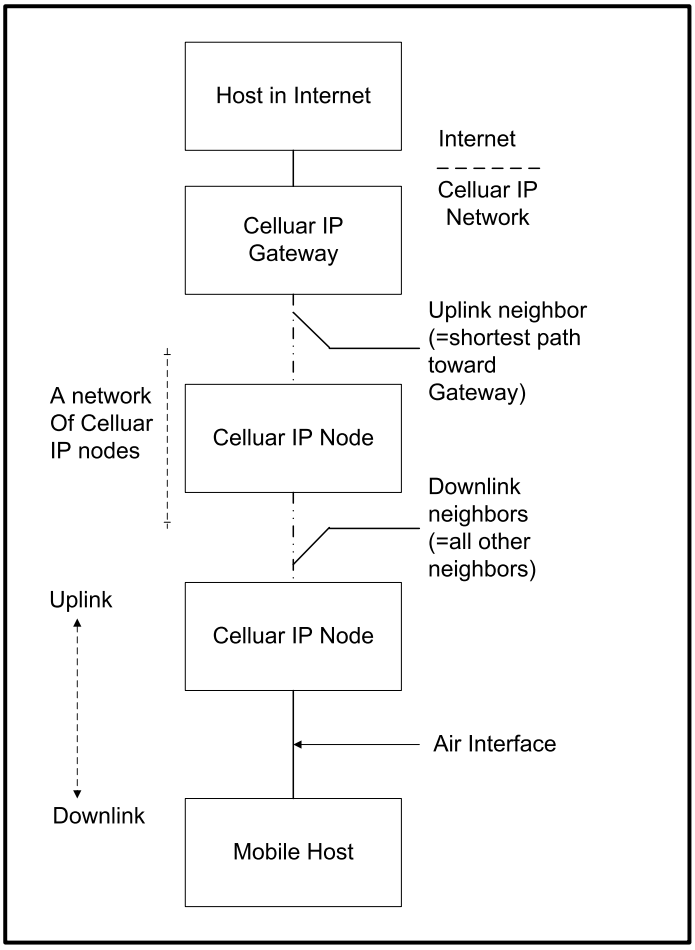

Abbildung 2.7: Lokalisierung und Routing in Celluar IP

Routing Wie bereits ausgeführt werden Pakete der Moble zum Gateway „hop-by-hop“ mit kürzester Pfadwahl geroutet. Die Celluar IP Knoten erstellen daraus ein Route Cache Mapping. Grundsätzlich sind Routing-Operationen denen des Lokalisierungs-Management gleich. Tabelle 2.1 [9, S. 8] zeigt Gemeinsamkeiten und Unterschiede von Paging und Route Cache in Bezug auf die gesendeten Steuerpakete.

	Paging Cache	**Route Cache**
Refresh erfolgt durch	Alle Uplink Pakete (Daten, Paging- und Route-Update)	Daten und Route-Update Pakete
Update durch	Alle Update-Pakete (Paging Update, Route Update)	Route-Update Pakete
Update-Zeitpunkt	Bewegung in neue Paging Area, Ablauf der Paging-Update Time	Bewegung in neue Zelle, Ablauf der Route-Update Time
Betroffene Mobile	Aktive und inaktive Mobile	Aktive Mobile
Zweck	Route Downlink Paket, falls kein Route Cache Eintrag	Route Downlink Paket

Tabelle 2.1: Vergleich Routing-Paging

Hard Handoff Ein Hard Handoff wird immer durch den Mobile Node ausgelöst. Sobald dieser sich an der neuen Basisstation anmeldet, überträgt die Mobile an diese ein Route Update Paket. Dieses Paket wird von der alten an die neue Basistation weitergeleitet. Im Route Cache und Paging Cache wird ein neues Mapping erzeugt. Analog dazu überträgt eine inaktive Mobile ein Route-Update Paket nur dann, wenn sie in eine neue Paging Area wechselt. Innerhalb der selben Paging Area wird kein Paket gesendet. Abbildung 2.8 verdeutlicht den Vorgang.

Semi-Soft Handoff Paketverluste während eines Hard Handoffs beeinträchtigen besonders bei TCP den Datendurchsatz. Der Durchsatz wird durch Verwendung des Semi-Soft Handoff verbessert. Während dieses Handoff bleibt die Mobile mit alter und neuer Basisstation in Kontakt und empfängt von beiden Pakete bzw. sendet an beide Stationen. Ein Semi-Soft Handoff läuft wie folgt ab:
Eine sich bewegende Mobile sendet an die neue Basistation ein Route Update Paket,verbleibt jedoch in Kontakt mit der alten Basistation. Das S-Flag wird gesetzt, um

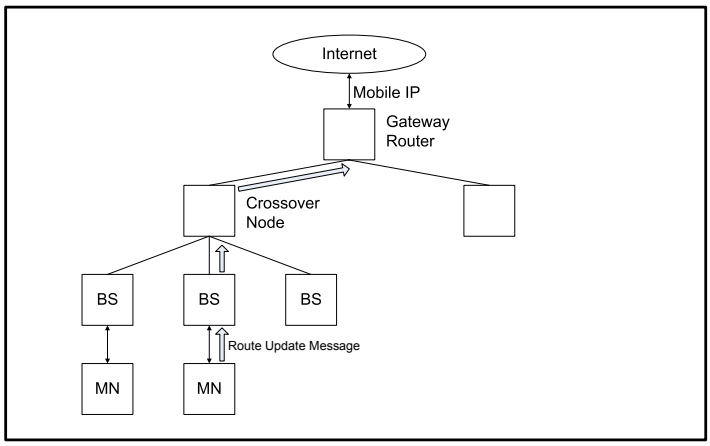

Abbildung 2.8: Handoff in Celluar IP

den Semi-Soft Handoff zu kennzeichnen. Im Cross-Over-Node wird das neue Mapping dem Cache hinzugefügt. Sobald ein zweites Route Update Paket an die neues Basisstation gesendet wird, ist das S-Flag nicht mehr gesetzt und das Mapping überschrieben. Somit ist der Semi-Soft Handoff bei minimaler Paketverlustrate abgeschlossen.

2.9.3 Zusammenfassung

Mittels Celluar IP wird die Mikromobilität verbessert. Das Lokalisierungs-Management ermöglicht eine verbesserte lokale Behandlung der Mobile innerhalb einer Domäne, besonders der Semi-Soft Handoff minimiert Paketverluste. Durch den Einsatz von Paging ist es möglich, Mobile in großen geografischen Regionen zu verwalten. Die getrennte Behandlung von aktiven und inaktiven Mobilen erzielt Performanzsteigerungen. Die Skalierbarkeit des Netzes, d. h. die Anzahl bedienter Mobile im Netz, kann erhöht werden. Routing erfolgt nur dann, wenn eine Mobile aktiv ist.
Celluar IP wirft ein Sicherheitsproblem auf. Route Update Pakete mit falscher Absenderadresse werden falsch weitergeleitet. Eine mögliche Lösung ist die Verwendung eines verschlüsselten Session Key, der zur Authentifizierung verwendet wird. Die Verwendung von IPSec erhöht zusätzlich die Sicherheit des Datenverkehrs.

2.10 HAWAII

Das Mikromobilitätsprotokoll *HAWAII* [10] steht für „Handoff-Aware Wireless Access Internet Infrastructure“ und bietet einen Domänen-basierten Ansatz zur Verbesserung

der Mikromobilität. Es setzt auf Mobile IP auf und verwendet lokal protokollspezifische HAWAII-Nachrichten, die den Verkehr zum Home Agent entlasten. Das Protokoll wurde 1999 von Lucent entwickelt und bei der IETF eingereicht. HAWAII verwendet spezialisierte Pfad Setup Schemen, die unter anderem Handoff-Vorgänge verwalten. Dazu werden Host-basierte Weiterleitungseinträge in spezifischen Routern installiert. Bei lokalen Operationen reduziert sich die Zahl der Mobilitätsaktualisierungen. Außerdem behalten Mobile während einer Bewegung innerhalb der Domäne ihre Netzwerkadresse, was bereits ein einfaches Quality of Service (QoS) darstellt. Desweiteren wird durch die Wartung von Soft-State Weiterleitungseinträgen für die Mobilen und Fehlererkennungsmechanismen bereits vorhandener Routing-Protokolle Zuverlässigkeit und Robustheit erreicht. HAWAII verfolgt folgende Entwurfsziele zur Verbesserung der Mikromobilität [10, S. 4]:

- Erzielen einer guten Performanz durch: Reduktion des Aktualisierungs-Verkehrs zum Home Agent, Vermeiden von Triangle Routing, Begrenzung der Störung des Benutzerverkehrs
- Intrinsische QoS-Unterstützung für das Mobilitätsmanagement
- Erhöhte Zuverlässigkeit und Verbesserung der Robustheit durch zusätzliche Protokoll-Mechanismen

2.10.1 Architektur

Ein weit verbreiteter Ansatz für die transparente Mobilität ist die Einteilung des Netzwerkes in Hierarchien. Bei HAWAII basiert die Hierarchie auf Domänen. Der Gateway zu jeder Domain wird „Domain Root Router“ genannt. Jede Mobile innerhalb der Domäne verfügt über eine IP-Adresse und korrespondiert zu einem Heimatnetz. Die Mobile behält diese IP-Adresse, solange sie sich nur innerhalb der Domäne bewegt. Datenpakete werden mittels der Subnet-Adresse des Domain Root Routers weiterleitet, was die Abdeckung geografisch großer Gebiete ermöglicht. In einer fremden Domäne erhält die Mobile zur lokalen Verwaltung eine Co-located Care-of Address (CCoA), die ursprüngliche Care-of Address (CoA) bleibt unverändert. Das vereinfacht das *per flow QoS* [10, S. 22]. Ein entscheidender Nachteil dieser Lösung ist die Verwendung von zwei IP-Adressen (CoA und CCoA) für eine Mobile. Die Lösung ist die Implementierung eines Dial-Up Modells für Drahtlosnetzwerke,wie sie von Internet Service Providern (ISP) verwendet wird. Abbildung 2.9 [10, S. 6] zeigt den Aufbau der HAWAII-Architektur.

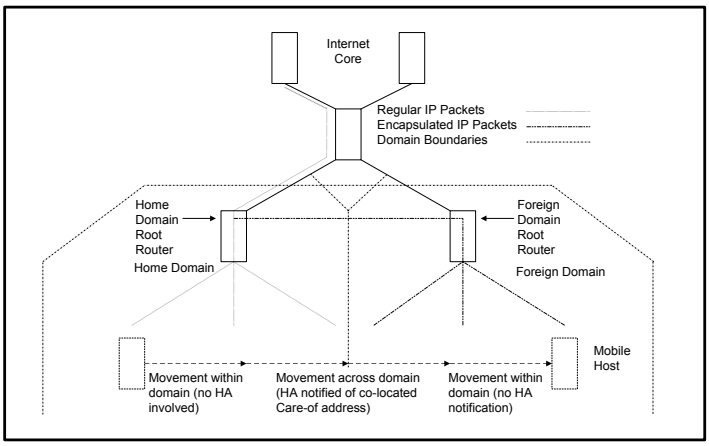

Abbildung 2.9: HAWAII-Architektur

2.10.2 Pfad Setup Schemen

HAWAII vergibt für jede Mobile eine eindeutige Adresse. Die Mobilitätsverwaltung dieser Ende-zu-Ende Konnektivität erfordert spezielle Techniken. Um Host-basierte Einträge in ausgewählten Routern vornehmen und aktualisieren zu können, werden Pfad Setup Update und Pfad Setup Refresh Nachrichten eingesetzt.

Pfad Setup Power Up Um sich im aktuellen Netzwerk zu registrieren, sendet die Mobile als erstes einen Mobile IP Registration Request an die nächste erreichbare Bassistation aus.
Diese sendet daraufhin eine Pfad Setup Update Nachricht zum Domain Root Router. Jeder Pfad-Router trägt einen Weiterleitungseintrag ein. Der Domain Root Router sendet ein Acknowledgement (Ack) zur Basisstation zurück, welche darauf einen Registration Reply an die Mobile sendet. Damit ist die Mobile im aktuellen Netz an der aktuellen Basisstation angemeldet; Abbildung 2.10 verdeutlicht die Zusammenhänge.

2.10.3 Paging

In HAWAII werden mehrere Basisstationen zu einer Paging Area zusammengefasst; genauer wird es es in der HAWAII-Protokollspezifikation [10] nicht definiert. Unterschieden werden hierarchische, personalisierte und feste Paging Areas.
Zwischen Mobile und Basisisstation verwaltet HAWAII das Paging. Jeder Paging Area wird eine Multicast-Adresse zugewiesen. Das benötigt weniger Bandbreite als eine

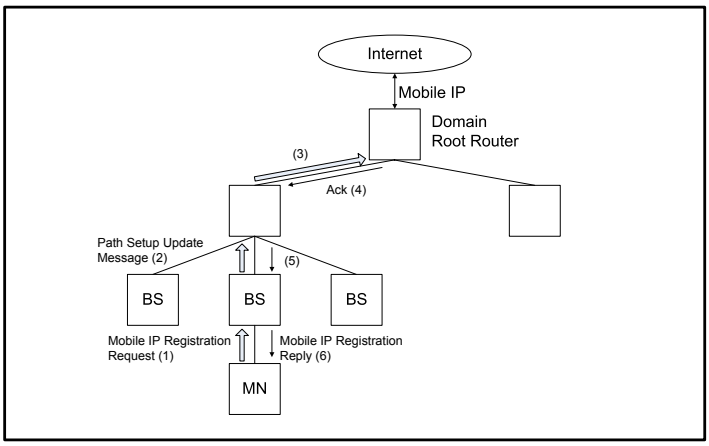

Abbildung 2.10: Pfad Setup Power Up

Unicast-Adresse. Paging Requests werden über das Multicast Routing Protokoll auf einem speziellen Paging Channel gesendet. Die Basisstation sendet über den Broadcast Channel regelmäßig Signale mit der ID der Paging Area aus. Datenpakete werden zwischengespeichert und eine Extrapaging Nachricht ausgesendet.

2.10.4 Forwarding Pfad Setup Schema

Bei dem Forwarding Pfad Setup Schema [10, S. 10], welches eine Form des Handoff bei HAWAII darstellt, werden die Pakete von der alten zur neuen Basisstation umgegeleitet, bevor sie den Cross-Over-Node erreichen. Es wird für Mobile verwendet, die nur an einer Basisstation angemeldet sein können. Abbildung 2.11 illustriert den genauen Ablauf der Prozedur.
Zuerst sendet die Mobile einen Mobile IP Registration Request (Nachricht 1) zu der neuen Basisstation. Diese Nachricht enthält die Adresse der alten Basisstation als Teil der Previous Foreign Agent Notification Extension (PFANE). Die neue Basisstation sendet eine Pfad Setup Update Nachricht (Nachricht 2) an die alte Basisstation. Die alte Basisstation führt einen Routing Table Lookup für die neue Basisstation durch. Über die Schnittstelle A werden die Pakete an Router 1 (Nachricht 3) weitergeleitet. Dazu wird ein Weiterleitungseintrag mit der IP-Adresse der Mobile im Router eingetragen. Die gleiche Operation wird durch Router 1 durchgeführt, der die Nachricht (Nachricht 4) an den Cross-Over-Router 0 überträgt. Dieser nimmt die Weiterleitungseinträge so vor, so dass die Mobile durch die neue Basisstation bedient wird. Die neue Basisstation ändert ihre Weiterleitungseinträge entsprechend um und sendet einen Mo-

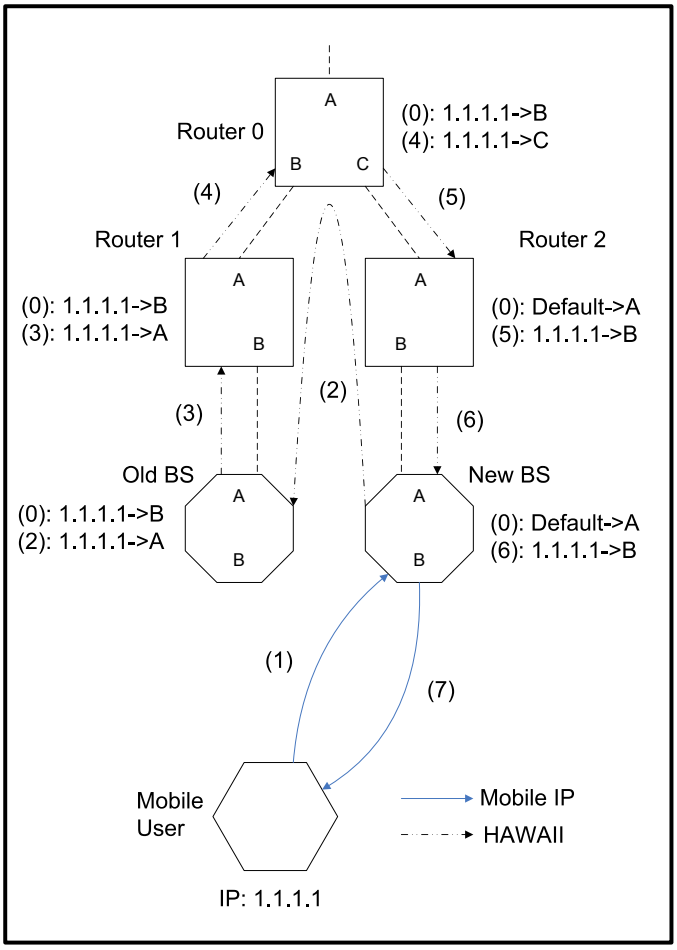

Abbildung 2.11: Forwarding Pfad Setup Schema

bile IP Registration Reply (Nachricht 7) an die Mobile. Damit ist der Handoff-Vorgang abgeschlossen.
Nur die neue und die alte Basisstation ist an dem Vorgang beteiligt. Somit erhalten nur der Domain Root Router und die Router auf dem Pfad zur neuen Basisstation periodische Refresh Nachrichten zur Aktualisierung der Routereinträge.

2.10.5 Non Forwarding Pfad Setup Schema

Das Non Forwarding Pfad Setup Schema [10, S. 10] ist eine weitere Form des Handoff bei HAWAII.

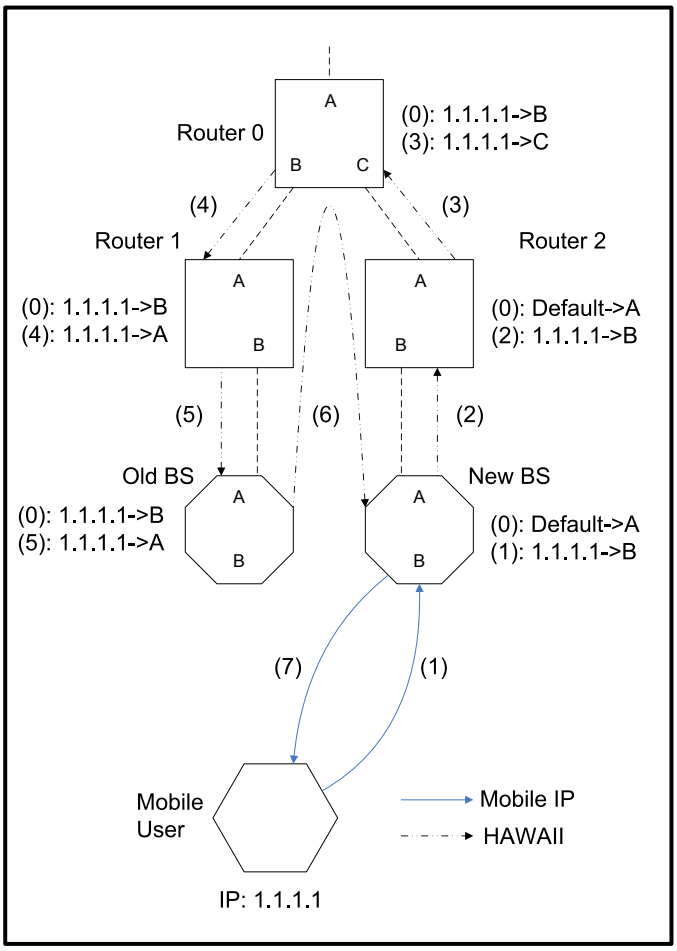

Abbildung 2.12: Non Forwarding Pfad Setup Schema

Im Gegensatz zum Forwarding Pfad Setup Schema werden die Pfad Setup Nachrichten

von der neuen zur alten Basisstation umgeleitet. Das Schema wird für Mobile verwendet, die mit mehr als einer Basisstation verbunden sein können. Der Cross-Over-Router leitet keine Pakete von der alten Basisstation weiter. Abbildung 2.12 zeigt den genauen Ablauf. Als erstes sendet die Mobile einen Mobile IP Registration Request (Nachricht 1) mit der PPFANE-Erweiterung aus. Die neue Basisstation nimmt einen Weiterleitungseintrag mit der IP-Adresse der Mobile vor. Dann wird ein Routing Table Lookup für die alte Basisstation durchgeführt. Die Pfad Setup Update Nachricht wird an Router 2 (Nachricht 2) und zum Cross-Over-Router 0 (Nachricht 3) weitergeleitet. Im Router 0 werden die Weiterleitungseinträge verändert, so dass die Pakete nur noch von der neuen Basisstation gesendet und empfangen werden können. In einigen Fällen erreicht die Pfad Setup Nachricht (Nachricht 5) noch die alte Basisstation. Diese ändert ihren Weiterleitungseintrag und sendet ein Acknowledgement (Ack, Nachricht 6) der Pfad Setup Nachricht an die neue Basisstation. Als letzter Schritt der Anmeldeprozedur sendet die neue Basisstation einen Mobile IP Registration Reply (Nachricht 7) an die Mobile.

2.10.6 Zusammenfassung

Mittels des Mobilitätsprotokolls HAWAII wird die Mikromoblität in drahtlosen Netzwerken verbessert. Als Erweiterung von Mobile IP bleibt die Makromobilität gewährleistet. Innerhalb einer Domäne wird durch HAWAII-spezifische Nachrichten der Verkehr zum Home Agent entlastet. Zudem sind über Pfad Setup Forwarding Schemen flexible Handoff-Vorgänge möglich. Soft-State Zustände in den Routern und regelmäßige Pfad Setp Refresh Nachrichten sorgen für einen hohen Aktualitätsgrad des Netzes. In HAWAII wird eine hohe Skalierbarkeit des Netzes und Robustheit gewährleistet. Zur Sicherheit erfolgt die Authentifizierung der Mobile bei Vergabe der co-located Care-of Address (CCoA). Nachteilig ist, dass in der HAWAII-Spezifikation Paging nicht genauer definiert ist.

2.11 Vergleich der untersuchten Protokolle

Die untersuchten Mikromobilitätsprotokolle Mobile IP, Hierarchical Mobile IP, Celluar IP und HAWAII arbeiten alle in der IP-Version 4. Alle drei Protokolle verwenden innerhalb einer Domäne hierarchische Mobilität, d. h. spezialisierte Router zur Verwaltung der lokalen mobilen Endgeräte. Bei Celluar IP ist es der Gateway Router, bei HMIP die Gateway/Regional Foreign Agents und bei HAWAII der Domain Root Router. Weitere

Gemeinsamkeiten sind schnelle Handoff-Techniken und Paging. Celluar IP unterstützt hierbei Hard- und Semisoft-Handoffs, bei HAWAII erfolgt der Handoff über Forwarding bzw. Non Forwarding Schemen. HMIP verwendet hierarchisches Tunneln im Gegensatz zu CIP und HAWAII, die lokal ein spezifisches Routing über protokollspezifische Nachrichten organisieren. CIP und HMIP verfügen zusätzlich über ein Sicherheitsmodell für den schnellen Handoff. Tabelle 2.2 [12] verschafft einen Überblick über die wichtigsten Eigenschaften.

	CIP	**HAWAII**	**HMIP**
OSI Layer	L3	L3	L3.5
Nodes Involved	all CIP nodes	all routers	FAs
Mobile Host ID	home addr	c/o addr	home addr
Intermediate Nodes	L2 switches	L2 switches	L3 routers
Means of Update	data pkt	signalling msg	signalling msg
Paging	implicit	explicit	explicit
Tunneling	no	no	yes
L2 Triggered Handoff	optional	optional	no
MIP Messaging	no	yes	yes

Tabelle 2.2: Vergleich der CIMS-Protokolle

3 Beschreibung von MIFA und L-MIFA

In diesem Kapitel sollen die beiden verbesserten Protokolle MIFA und L-MIFA in ihrer Funktionsweise vorgestellt werden. MIFA steht für „Mobile IP Fast Authentication Protocol“ und ist eine Entwicklung des Fachgebiets IHS der TU Ilmenau. Duch diese Protokoll sollen Schwächen des MIP-Protokolls beseitigt werden ohne Einschränkungen am verwendeten Netzwerk vornehmen zu müssen.
L-MIFA steht für „Low Latency Mobile IP Fast Authentication Protocol“ und ist einer Weiterentwicklung von MIFA. L-MIFA ermöglicht nahtlose schnelle Handoff-Vorgänge.

3.1 MIFA

Die Untersuchungen haben gezeigt, dass zur Verbesserung der Mikromobilität ein hoher Aufwand betrieben werden muss. Es werden innerhalb des Netzwerkes Router mit Spezialaufgaben benötigt, was den Organisationsaufwand für das Netzwerk erhöht. Genau hier liegt der Ansatz für die Entwicklung eines neuen Mobilitätsprotokolls, dass keine komplexe Infrastruktur und Router mit Spezialaufgaben benötigt.
Das Mobilitätsprotokoll MIFA [13] ist entwickelt wurden, um Handoff-Vorgänge in Frequent Handoff Regions (FHR) zu beschleunigen. Dies führt zur Verbesserung der Dienstgüte von Echtzeitanwendungen in kleinen Funkzellen, beispielsweise WLAN. Desweiteren werden Sicherheitsaspekte bei der Übertragung betrachtet. Die Grundidee bei MIFA ist die Delegation der lokalen Authentifizierung der Mobile im neuen Subnetz vom Home Agent an den Foreign Agent. Der vorhergehende Foreign Agent wird über die neue Verbindung informiert. Über den vorhergehenden Foreign Agent werden die Datenpakete während des Handoff getunnelt, bis der Home Agent informiert und der Handoff abgeschlossen ist.
Betrachtet werden die Registrierungsprozeduren in MIFA. Des weiteren wird die SDL-Spezifikation des MIFA-Protokolls vorgestellt, um die Operationen in MIFA zu verdeutlichen.

3.1.1 Registrierung in MIFA

Die lokale Authentifizierung setzt eine Gruppe benachbarter Foreign Agents vorraus, die eine *Layer-3 Frequent Handoff Region* (L3-FHR) bilden. Die L3-FHR besteht typischerweise aus einer kleinen Anzahl Foreign Agents, in der Mobile sich bewegen und häufig den FA wechseln. Diese Regionen können statisch aufgebaut sein, dynamisch durch das Netz selbst oder durch Standard Algorithmen gebildet werden, z. B. durch den *Neighbor Graph Algorithmus* [15]. In Abbildung 3.1 [13, S. 3], die den Handoff-Vorgang bei MIFA verdeutlicht, bilden vorhergehender und aktueller FA je eine L3-FHR.

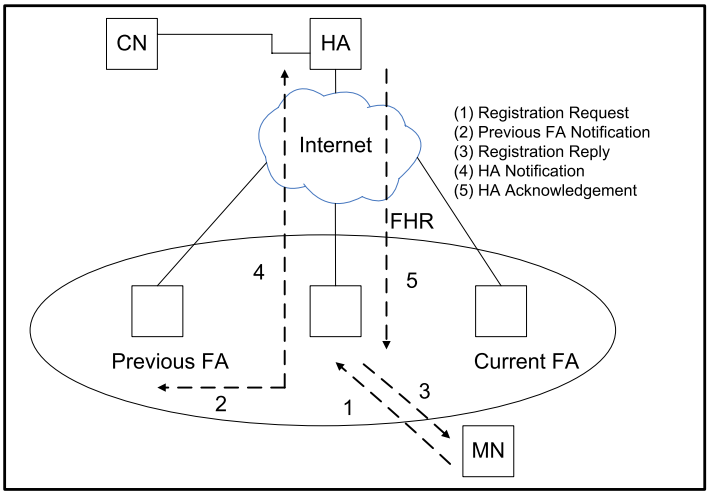

Abbildung 3.1: Registrierung in MIFA

Zwischen den Foreign Agents einer L3-FHR bestehen Security Associations, die durch den Netzwerkadministrator oder das Netzwerk selbst gebildet wird. Wenn eine Mobile sich in ein neues Subnetz bewegt und den Link zum alten FA verliert, sendet sie als erstes einen Registration Request (MIFA_REG_REQUEST) (1) an den neuen Foreign Agent. Der vorhergehende Foreign Agent erhält einen Previous FA Notification (P_FA_NOT) (2). Nachdem der aktuelle FA mit einem Registration Reply (MIFA_REG_REPLY) (3) den Registration Request der Mobile beantwortet hat, kann die Übertragung wieder aufgenommen werden. Bei einer Downlinkverbindung werden dazu die Pakete über den vorhergehenden FA an den aktuellen FA weitergeleitet. Mit der Information an den Home Agent, der HA Notification (HA_NOT) (4), wird der Handoff fortgesetzt. Abgeschlossen wird der Vorgang mit dem Erhalt des Previous FA Acknowledgement (P_FA_Ack) und dem HA Acknowledgement (HA_Ack) (5) durch den aktuellen FA. Damit der ist der Handoff abgeschlossen. Der Tunnel zum vorherge-

henden FA bei Downlinkverbindungen wird aufgehoben. Die Pakete werden nur noch über den aktuellen FA weitergeleitet.

3.1.2 Spezifikation des Protokolls

In diesem Abschnitt wird des MIFA-Protokoll im Detail beschrieben. Dazu werden alle Prozeduren ausführlich erläutert. Der verbesserte Handoff-Vorgang soll damit verständlich gemacht werden.

Initial Authentication Exchange und Security Association MIFA verwendet Mobile IP Operationen zur Initialisierung der Registrierung. Zusammen mit dem Registration Request wird ein Flag übertragen, das die Verwendung von MIFA für künftige Operationen kennzeichnet. Der Home Agent bildet einerseits die Security Association ($K1_{FA,HA}$) zwischen HA und FA und andererseits die Security Association ($K1_{MN,FA}$) zwischen MN und FA. Als nächsten Schritt führt der Foreign Agent die *Initial Authentication Exchange Prozedur* durch. Der aktuelle FA sendet eine *Move Probability Notification Message* (M_P_Not) an den Home Agent. Diese Nachricht enthält die zwei Zufallsvariablen $R1$ und $R2$ und die Security Association ($K2_{FA,HA}$) zwischen dem Home Agent und aller FAs in der L3-FHR. ($K2_{FA,HA}$) wird mittels ($K1_{MN,FA}$) verschlüsselt. Durch diesen Schlüssel ist die Nachricht geschützt. Der Home Agent beantwortet die Notifikation durch Senden des *Move Probability Acknowledgement* (M_P_Ack) an den aktuellen FA. Abbildung 3.2 [13, S. 4] zeigt den Ablauf der Initial Authentication Exchange Prozedur.

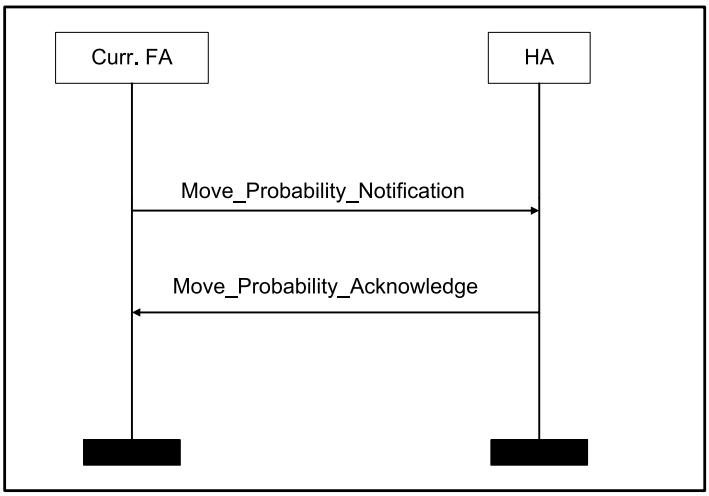

Abbildung 3.2: Initial Authentication Exchange

Move Notification Mittels der *Move Notification Prozedur* wird die Möglichkeit des Handoffs einer Mobile geprüft. Dazu sendet der aktuelle FA ein Move Probability Notification Nachrichten (M_P_NOT) an alle benachbarten Foreign Agents der L3-FHR. Diese Nachrichten enthalten die Security Association ($K2_{MN,FA}$) zwischen MN und den FAs der L3-FHR und die Security Association ($K2_{FA,HA}$) zwischen diesen Foreign Agents und dem Home Agent. Diese Security Associations werden mittels der Shared Security Association ($K_{FA,FA}$) zwischen den einzelnen FAs verschlüsselt, worüber die Move Probability Notification Nachrichten authentifiziert werden. Wenn sich nun die Mobile zu einem der FAs bewegt, antwortet dieser mit einer Move Probability Acknowledgement (M_P_Ack) dem aktuellen FA. Abbildung 3.3 [13, S. 4] verdeutlicht den Ablauf der Prozedur.

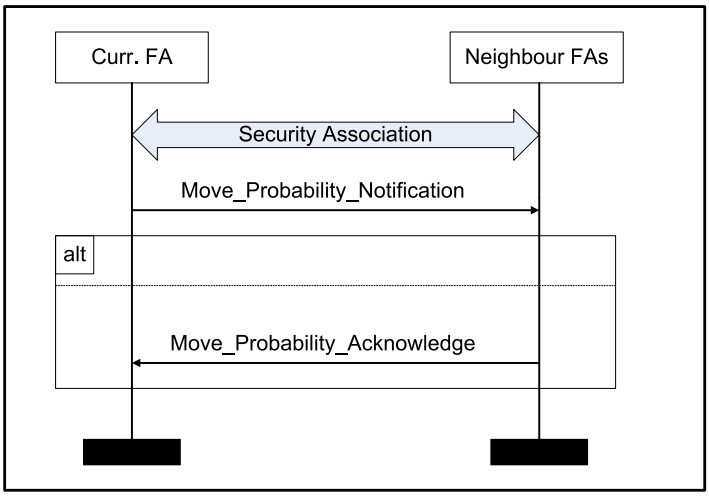

Abbildung 3.3: Move Notification

Authenticators Exchange Mit Hilfe der *Authenticators Exchange Prozedur* wird die Authentifizierungsinformation an jeden Foreign Agent der L3-FHR übertragen. Jeder Foreign Agent sendet eine Authentication Information Request Nachricht (Auth_Inf-_Rqst) an den Home Agent, welche durch ($K2_{FA,HA}$) authentifiziert wird. Der Home Agent antwortet mit einer Authentication Information Response Nachricht (Auth_Inf-_Response). Diese Nachricht enthält Authentifizierungsinformationen zwischen MN und HA, um zukünftige Registrierungen vornehmen zu können. Diese Nachricht ist ebenfalls durch ($K2_{FA,HA}$) authentifiziert.
Die Authenticators Exchange Prozedur ist ein zusätzliches Feature, denn Informationen der Authentication Information Response Nachricht können auch mit der Move

Probability Acknowledgement Nachricht vom Home Agent zum aktuellen FA gesendet werden. Dies erfolgt durch die Initial Authentication Exchange Prozedur. Der aktuelle FA verteilt diese Informationen an die benachbarten Foreign Agents und garantiert so die Skalierbarkeit des MIFA-Protokolls. In der aktuellen MIFA-Implementierung wird diese Prozedur nicht verwendet. Abbildung 3.4 [13, S. 5] zeigt den Nachrichtenfluss der Prozedur.

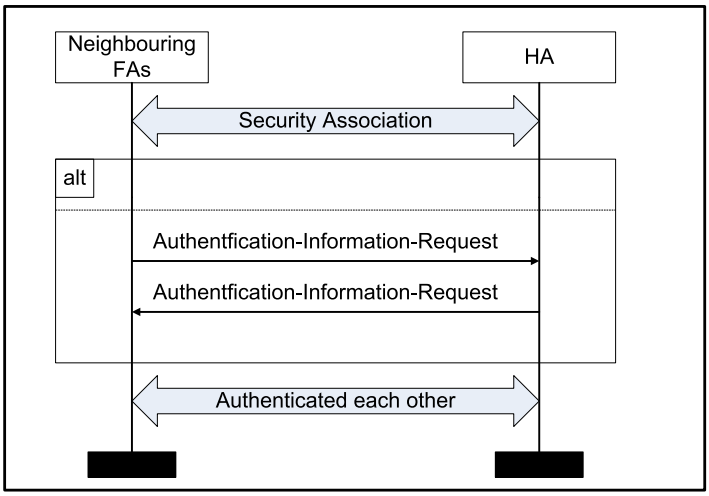

Abbildung 3.4: Authenticators Exchange

Registration by Neighbour Agent Wenn eine Mobile sich in der L3-FHR bewegt, wird die *Registration by Neighbour Agent Prozedur* durchgeführt. Der MN erhält vom aktuellen FA eine Agent Advertisement Nachricht (MIPT_ADS). Der MN sendet daraufhin den Registration Request (MIFA_REG_REQUEST) an diesen Foreign Agent, welcher mittels ($K2_{MN,FA}$) authentifiziert wird. Nach erfolgreicher Authentifizierung sendet der aktuelle FA die Previous Foreign Agent Notification Nachricht (P_FA_NOT) an den vorhergehenden Foreign Agent. Diese Nachricht ist durch die existierende Security Association zwischen den Foreign Agents der L3-FHR authentifiziert. Dadurch wird das Tunneln von Datenpaketen im Downlink an die Mobile über den vorhergehenden FA an den aktuellen FA sichergestellt. Der vorhergehende FA antwortet mit einer Move Probability Acknowledgement Nachricht (M_P_Ack) und beginnt die Pakete zu tunneln.
Jetzt generiert der aktuelle FA den Schlüssel ($K3_{MN,FA}$) für die Authentifizierung des MN und des nächsten FA in einer neuen L3-FHR. ($K3_{MN,FA}$) wird mittels ($K2_{MN,FA}$) verschlüsselt. Der aktuelle Fa erzeugt drauf hin zwei Zufallsvariablen $R1$ und $R2$ und

schickt einen Registration Reply (MIFA_REG_REPLY) an den MN. Anschließend erzeugt der aktuelle FA den Schlüssel ($K3_{FA,HA}$), der mittels ($K2_{FA,HA}$) verschlüsselt wird, zur Authentifizierung zwischen dem aktuellen FA und HA in einer neuen L3-FHR. Dann informiert der aktuelle FA den Home Agent mittels der Home Agent Notification Nachricht (HA_NOT) über die neue Verbindung. Der Home Agent sendet nach Erhalt der Notification Nachricht die Nome Agent Acknowledgement Nachricht (HA_Ack) an den aktuellen FA und errichtet damit dauerhaft die Verbindung zum aktuellen FA, wobei der Tunnel zum vorhergehenden FA aufgehoben wird. Der Home Agent sendet Authentifizierungsinformationen, die für die nächste Registrierung der Mobile benötigt werden, mit dem nächsten HA_Ack an den nächsten Foreign Agent. Der neue Foreign Agent informiert benachbarte Foreign Agents mittels der Move Notification Prozedur über eine mögliche Bewegung der Mobile. Anschließend startet der MN erneut die Registration by Neighbour Agent Prozedur. Ab diesem Zeitpunkt wiederholt sich der beschriebene Vorgang, Abbildung 3.5 [13, S. 6] verschafft einen Überblick.
Die Abwärtskompatibilität zu Mobile IP soll gewahrt bleiben,um Funktionsweise des Netzwerkes bei eventuell auftretenden Fehlern zu gewährleisten. Dazu enthält die MIFA Registration Request Nachricht (MIFA_REG_REQUEST) eine Erweiterung mit der Authentifizierungsinformation zwischen HA und MN. Sobald ein Fehler in einer der MIFA-Prozeduren auftritt oder auch nur eine MIFA-NAchricht verloren geht, wird Mobile IP verwendet.

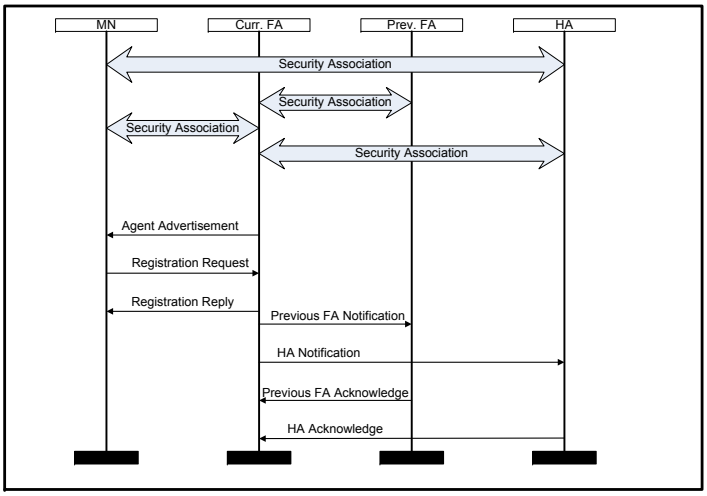

Abbildung 3.5: Registration by Neighbour Agent

3.1.3 Operationen in MIFA

Der folgende Abschnitt beschreibt anhand der SDL-Spezifikation des MIFA-Protokolls, dargestellt in Abbildung 3.6 [13, S. 7], den genauen Aufruf der MIFA-Prozeduren sowie auftretende Layer-2 und Layer-3 Handoffs.

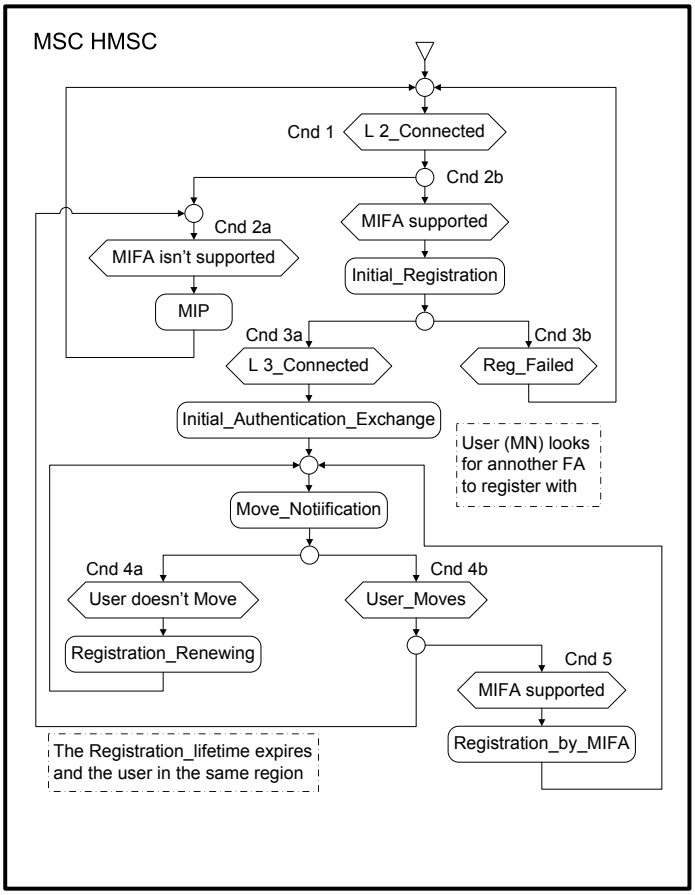

Abbildung 3.6: Operationen in MIFA

Sobald sich die Qualität der Verbindung zwischen aktuellen Foreign Agent und Mobile Node verschlechtert, scannt die Mobile die Umgebung nach verfügbaren Access Points (AP). Ist ein passender AP gefunden, authentifiziert und verbindet sich die Mobile mit diesem Access Point. Damit ist zunächst eine Layer-2 Verbindung hergestellt (Cnd 1). Falls der neue AP zu einem andern Subnetz gehört als der alte Access Point, muss die Mobile den dazu zugehörigen Foreign Agent finden. Falls der neue Foreign Agent

MIFA nicht unterstützt (Cnd 2a), registriert sich die Mobile beim Home Agent unter Verwendung von Standard Mobile IP. Wenn der Link zu diesem AP verloren geht, kehrt der MN zu Cnd 1 zurück. Falls der neue FA MIFA unterstützt (Cnd 2b), führt der MN die initiale Registrierungsprozedur durch. Schlägt diese Prozedur fehl, scannt der Mobile Node erneut nach verfügbaren Access Points und kehrt zu Cnd 1 zurück. Bei erfolgreicher Initialisierung der Mobile ist eine Layer-3 Verbindung (Cnd 3a) hergestellt. Danach werden nacheinander die MIFA-Prozeduren *Initial Authentication Exchange* (siehe Abschnitt 3.1.2) und *Move Notification* (siehe Abschnitt 3.1.2) durchgeführt. Bei zu langsamer Bewegung der Mobile läuft die *Registration Lifetime* ab, bevor diese sich in ein neues Subnetz (Cnd 4a) bewegt hat. In diesem Fall wird die Registrierung erneut durchgeführt und die Move Notification Prozedur wiederholt. Wenn die Mobile sich vor Ablauf der Registration Lifetime in neues Subnetz bewegt (Cnd 4b), registriert sich die Mobile bei dem neuen Foreign Agent entweder mittels MIFA (Cnd 5) oder Mobile IP (Cnd 2a).

3.2 Das Protokoll L-MIFA

L-MIFA [17] steht für „Low Latency Mobile IP Fast Authentification Protocol“ und erweitert MIFA um schnelle und nahtlose L2-Handoffs. Im folgenden Abschnitt soll die Arbeitsweise dieses Protokolls beschrieben werden.

3.2.1 Initiale Registrierung in L-MIFA

L-MIFA [17, S. 2] erweitert das Protokoll MIFA um Funktionen, die Informationen des Layer-2 nutzen. Ein Layer-2 Trigger zeigt die Bewegung der Mobile in eine neue Zelle an. Für die initiale Registrierung nutzt die Mobile das Mobile IP Protokoll. Zusätzlich informiert die Mobile den Home Agent und den Foreign Agent über die Verwendung des L-MIFA Protokolls für zukünftige Operationen. Mittels eines Flags in der *Registration Request Nachricht* wird die Verwendung von L-MIFA gekennzeichnet. Durch den Home Agent wird die Security Association $K1_{FA,HA}$ zwischen Home Agent und Foreign Agent sowie die Security Association $K1_{MN,FA}$ zwischen dem Mobile Node und dem Foreign Agent gebildet. Zusätzlich generiert der Home Agent die zwei Zufallsvariablen $R1$ und $R2$ sowie den weiteren Schlüssel $K2_{MN,FA}$ zur Authentifizierung zwischen Mobile und Foreigen Agent.
$K1_{MN,FA}$, $K2_{MN,FA}$ sowie $R1$ und $R2$ werden zusammen mit der *Registration Reply Nachricht* zur Mobile gesendet. Der Schlüssel $K2_{MN,FA}$ wird zur Authentifizierung der

Mobile am nächsten Foreign Agent verwendet. Als nächsten Schritt sendet der Foreign Agent, zu dem sich die Mobile als nächstes bewegt, die *Move Probability Notification Nachricht* zusammen mit dem Schlüssel $K2_{FA,HA}$, der durch $K1_{FA,HA}$ verschlüsselt ist, zum Home Agent. Dieser neue Foreign Agent gehört gewöhnlich zur selben L3-FHR wie der aktuelle Foreign Agent. Der Home Agent antwortet mit der *Move Probability Acknowledgement Nachricht*. Diese Nachricht enthält die Authentifizierung der Mobile mit dem nächsten Foreign Agent.
Grundsätzlich führt der Mobile Node diese MIFA-Registrierungsprozedur nur bei der erstmaligen Registrierung beim Home Agent durch. Alle weiteren Registrierungen erfolgen mittels des L-MIFA Protokolls.

3.2.2 Operationen in L-MIFA

Der Mobile Node scannt nach neuen verfügbaren Access Points, wenn sich die Qualität der Verbindung verschlechtert. Falls der neue Access Point zu einem Subnetz gehört, wird ein L2-Trigger ausgelöst, der zur Identifizierung des nächsten Foreign Agent genutzt wird. Die Mobile führt einen Layer-3 Handoff durch, der ebenfalls durch den L2-Trigger ausgelöst wird. Die Mobile sendet als erstes eine *Proxy Router Solicitation Nachricht* (PRSol) an den aktuellen Foreign Agent. Diese Nachricht basiert auf der *Agent Solicitation Nachricht* (AgSol) aus Mobile IP. Der aktuelle Foreign Agent beantwortet die Solicitation Nachricht mit der *Proxy Router Advertisement Nachricht* (PRAdv). Die Advertisement Nachricht enthält die Care-of Address des neuen Foreign Agent sowie ein Flag zur Kennzeichnung des L-MIFA Protokolls. Es wird vorrausgesetzt, dass Agent Advertisement und -Solicitation Nachrichten zwischen Foreign Agents der selben L3-FHR ausgetauscht werden.
Als nächstes sendet der Mobile Node die *Registration Request Nachricht*, die durch die Security Association $K2_{MN,FA}$ authentifiziert ist, an den aktuellen Foreign Agent. Anschließend sendet der aktuelle Foreign Agent die *Initial Acknowledgement Nachricht* (IntAck) zum Mobile Node. Daurch ist der Mobile Node über den korrekten Empfang des Registration Request informiert, d. h. dass die Request Nachricht nicht gedroppt wurde. Jetzt verschlüsselt der aktuelle Foreign Agent die Security Association $K2_{MN,FA}$ zwischen Mobile Node und dem neuen Foreign Agent und die Security Association $K2_{FA,HA}$ zwischen dem neuen Foreign Agent und dem Home Agent mittels der bestehenden Security Associations $K_{FA,FA}$ zwischen den Foreign Agents der L3-FHR. Diese Schlüssel werden der Registration Request Nachricht hinzugefügt. Der aktuelle Foreign Agent authentifiziert die Registration Request Nachricht mittels $K_{FA,FA}$ und

sendet diese zum neuen Foreign Agent. Abbildung 3.7 [17, S. 2] zeigt den Nachrichtenfluss in L-MIFA.

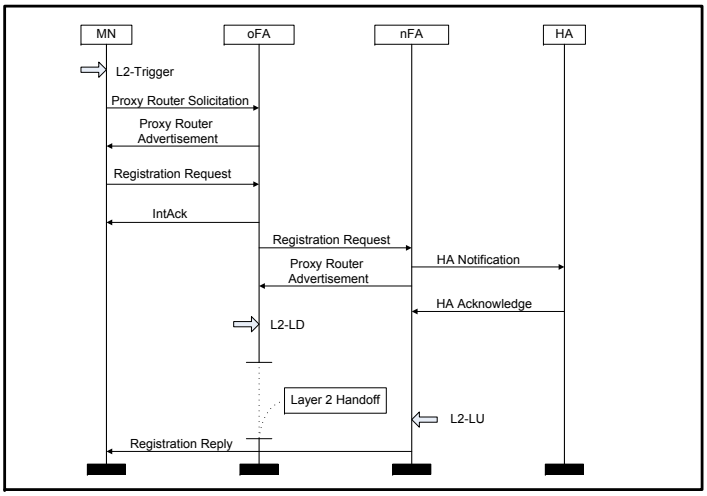

Abbildung 3.7: Registrierung mit L-MIFA

Nach Überprüfung der Authentifizierung zwischen den Foreign Agents mittels $K_{FA,FA}$ entschlüsselt der aktuelle Foreign Agent $K2_{MN,FA}$ und prüft die Authentifizierungsinformationen zwischen Mobile Node und Foreign Agent. Bei Übereinstimmung der berechneten Authentifizierungswerte durch Mobile und Foreign Agent informiert der aktuelle Foreign Agent mittels der *Previous Foreign Agent Notification Nachricht* (P-_FA_NOT) den vorhergehenden Foreign Agent. Somit werden die Pakete bei einer Downlinkverbindung vom Foreign Agent zum Mobile Node, ausgelöst durch den L2-Trigger *Layer-2 Link Down* (L2-LD), über den vorhergehenden Foreign Agent getunnelt. Die Previous Foreign Agent Notification Nachricht ist durch den $K_{FA,FA}$ authentifiziert. Der neue Foreign Agent generiert zwei neue Zufallsvariablen $R1$ und $R2$ sowie den Schlüssel $K3_{FA,HA}$, der mit $K2_{FA,HA}$ verschlüsselt wird. Mit diesen Werten werden die Nachrichten zwischen dem nächsten Foreign Agent und dem Home Agent authentifiziert, sobald die Mobile sich registriert. Der neue Foreign Agent sendet $R1$, $R2$ und $K3_{FA,HA}$ zusammen mit der *Home Agent Notification Nachricht* (HA_NOT), die durch den Schlüssel $K2_{FA,HA}$ authentifiziert wird, an den Home Agent.
Der Home Agent antwortet mit der *Home Agent Acknowledgement Nachricht* (HA-_Ack), die die Authentifizierungswerte der Mobile für die nächste Registrierung enthält. L-MIFA unterscheidet zwischen Up- und Downlinkverbindungen mit Hilfe von Layer-2 Triggersignalen. Bei Empfang des Triggersignals L2-LD tunnelt der vorhergehende Foreign Agent die Pakete zum neuen Foreign Agent. Empfängt der neue Foreign Agent

das Triggersignal *Layer-2 - Link Up* (L2-LU), so erzeugt dieser den durch $K2_{MN,FA}$ verschlüsselten Schlüssel $K3_{MN,FA}$ sowie die Zufallsvaraiblen $R1$ und $R2$ zur Authentifizierung zwischen Mobile und Foreigen Agent und sendet diese Daten zusammen mit der Registration Reply Nachricht an den Mobile Node. Mit Abschluss der Registrierung wird die Übertragung wieder aufgenommen.

3.3 Zusammenfassung

MIFA weist gegenüber den anderen untersuchten Mobilitätsprotokollen (CIP, HMIP, HAWAII) zahlreiche Vorteile auf. Das Protokoll bietet intrinsische Sicherheit der Verbindung. Diese wird durch Authentifizierung zwischen MN und FA mittels Security Associations erreicht. Diese bestehen zwischen Mobile Node und Home Agent sowie dem Mobile Node und dem Foreign Agent.
Ein entscheidender Vorteil von MIFA ist, dass es ohne hierarchische Foreign Agents bzw. spezialisierte Router auskommt. Lediglich Foreign Agent und Home Agent müssen MIFA unterstützen. Damit ist der Organisationsaufwand für das Netzwerk geringer. Ebenfalls bleibt die Abwärtskompatibilität zu dem Protokoll Mobile IP gewährleistet, was bei Fehlern die Funktion des Netzwerkes garantiert und robuster gegenüber Totalausfällen gestaltet.
MIFA hat sowohl Mikro- als auch Makromobilitätseigenschaften, es wird kein gesondertes Protokoll für die Makromobilität benötigt. Dies ist beispielsweise bei HAWAII oder HMIP der Fall, die für die Kommunikation mit dem Home Agent Mobile IP verwenden. Um den Handoff zwischen verschiedenen Domänen zu verbessern, wird HMIP und MIFA kombiniert. Dazu muss MIFA zusätzlich im GFA unterstützt werden. Dadurch kann die Mobile ohne Registrierung beim Home Agent die Domäne wechseln.
Durch die veränderte Form der Registrierung kann bei L-MIFA die Handoff Latenzzeit noch weiter verkürzt werden. Auch bei L-MIFA müssen keine Einschränkungen der Netzwerktopologie vorgenommen werden.

4 Der Netzwerksimulator ns-2

Der Netzwerksimulator NS-2 [4] ist ein ereignisorientierter Simulator, der in C++ und OTcl geschrieben ist. Entwickelt wurde er an der Universität Berkeley und ist in Open Source verfügbar. NS-2 ermöglicht es, die für die Arbeit erforderlichen Simulationen verdrahteter und drahtloser Netze mit verschiedenen Verkehrsquellen durchzuführen und auszuwerten. Durch Kombination von Tcl/Tk und C++ ergeben sich Vorteile. In C++ implementierte Änderungen können schnell ausgeführt, in Tcl/Tk können Änderungen schnell vorgenommen werden.
Sehr nachteilig wirkt sich die umständliche Handhabung des Simulators aus; da es sich um Open Source handelt, können Fehler und Bugfixes nur im Quelltext behandelt werden. Somit sind sichere Programmierkenntnisse in C++ und Tcl/Tk Vorraussetzung für die solide Handhabung. Abbildung 4.1 [3] zeigt den Aufbau des Simulators aus der Sicht des Nutzers. Weitere wichtige Bestandteile sind der Network Animator (Nam) und das Auswertungstool xgraph für Diagrammfunktionen, die in diesem Kapitel kurz vorgestellt werden sollen.

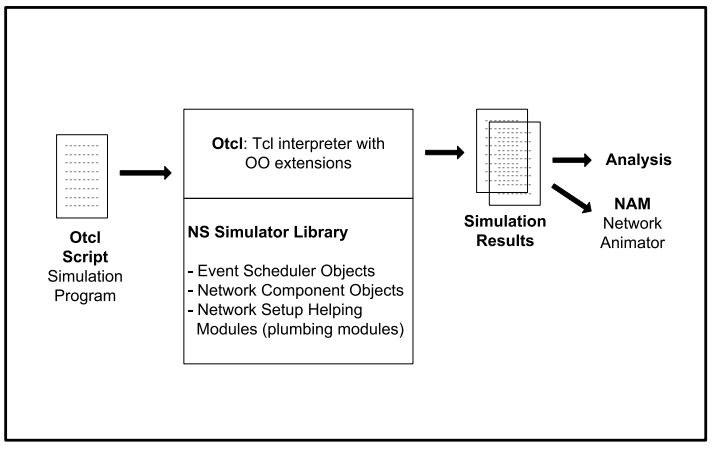

Abbildung 4.1: NS-2 aus der Sicht des Nutzers

4.1 Anwendungsgebiete des Netzwerksimulators

Im Netzwerksimulator ns-2 können komplexe Szenarien für die verschiedensten Netzwerke (UMTS, WLAN, usw.) getestet werden. Ebenfalls möglich ist die Implementierung neuer Protokolle, beispielsweise Routing- oder Mobilitätsprotokolle. Diese Protokolle können anschließend evaluiert und mit anderen Protokollen verglichen werden. Der Vorteil des Simulators liegt auf der Hand. Komplexe Netzwerke müssen zum Testen eines neuen Protokolls nicht real aufgebaut werden. Zum Testen der Perfomanz können spezielle Trace-Dateien erzeugt werden, die zur Auswertung dienen.

4.2 Der Network Animator Nam

Nam [4] ist neben xgraph das einzige grafische Visualisierungstool des ns-2 Paketes. Es ist Tcl/Tk basiert und dient zur Anzeige so genannter nam-Tracedateien. Diese Tracedateien werden aus Szenarien erzeugt, die in Tcl-Skripten abgelegt sind. Nam bietet verschiedene Monitoring-Tools zur Überwachung des Paketverkehrs und zur Kontrolle der richtigen Arbeitsweise von Protokollen. Das Aufstellen komplexer Topologien wird unterstützt, die zur besseren Übersicht beschriftet und farblich gestaltet werden können. Abbildung 4.2 zeigt eine typische Ausgabe von Nam. Der TCP-Verkehr zwischen zwei Knoten wird grafisch dargestellt, in der Warteschlange befindliche Pakete werden sichtbar gemacht.

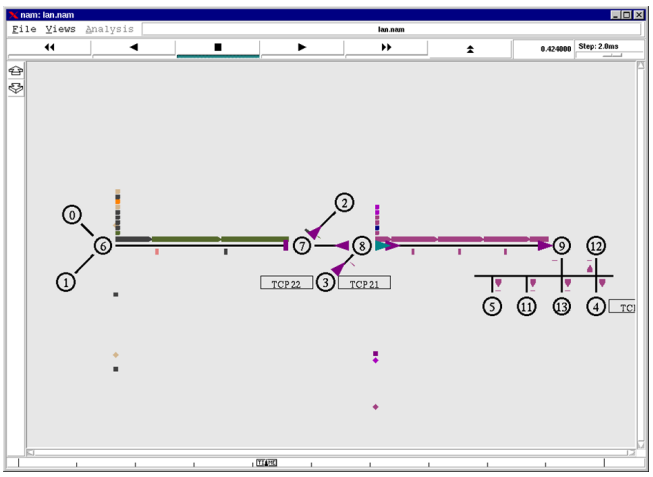

Abbildung 4.2: Der Network Animateur

4.3 Das Diagrammtool Xgraph

Mit Hilfe des Tools Xgraph [4] können aus speziell erzeugten Tracedateien interaktiv grafische Anzeigen beispielsweise über Paketverluste oder TCP-Durchsätze grafisch dargestellt werden. Damit steht in ns-2 ein schlankes Tool zur Verfügung, mit dem schnell und effektiv Statistiken zur Anzeige gebracht werden können. Abbildung 4.3 zeigt als Beispiel drei farbige Kurven des TCP-Durchsatzes.

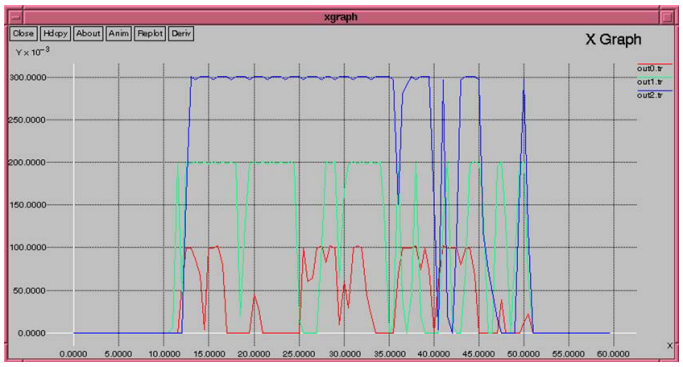

Abbildung 4.3: Xgraph

5 Implementierung von L-MIFA in ns-2

In diesem Kapitel wird die Vorgehensweise der Implementierung von L-MIFA in ns-2 beschrieben. Die wichtigsten Routinen, die sich im Anhang B im Quelltext befinden, werden erläutert. Zunächst wird die Funktion von L-MIFA noch einmal kurz erläutert. Als nächstes werden die dazu notwendigen Nachrichten definiert und beschrieben, wie diese in ns-2 aufgerufen werden. Zum Schluss werden alle notwendigen Prozeduren erläutert. Am Ende das Kapitels wird das Szenario erläutert, das für die Bewertung der untersuchten Protokolle verwendet wird.

5.1 Nachrichten in L-MIFA

Für die Nachrichten von L-MIFA werden geeignete Bezeichnungen gewählt, so dass diese in ns-2 eingebunden werden können. Abbildung 5.1 zeigt den Fluss der Nachrichten in L-MIFA. Alle in Klammern verwendeten Abkürzungen der Nachrichten werden so in ns-2 verwendet. Bei einem Handoff erhält die Mobile (MH) ein Advertisement Paket (ADV). Daraufhin sendet MH die Proxy Router Solicitation Nachricht (PRSOL) an den alten FA (oFA). Der oFA beantwortet diese mit einer Proxy Router Advertisement Nachricht (PRADV). MH sendet hierauf den Registration Request 1 (REGREQ1) an den oFA. Der oFA sendet nach Erhalt des REGREQ1 die Registration Request 2 Nachricht (REGREQ2) an den neuen FA (nFA). Der nFA informiert den HA mittels der Home Agent Notification Nachricht (HA_NOT) und erhält als Bestätigung die HA Acknowledge Nachricht (HA_ACK). Abschließend sendet der nFA die Registratin Reply 1 Nachricht (REGREPLY1) an den oFA und die Registratin Reply 2 Nachricht (REGREPLY2) an den MH.

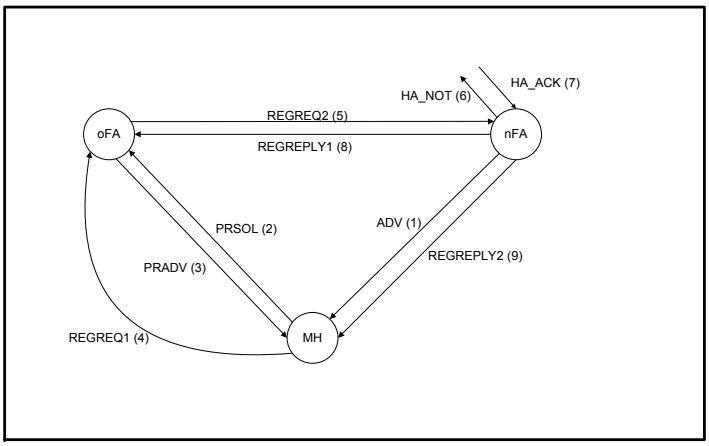

Abbildung 5.1: Nachrichtenfluss in L-MIFA

5.2 Beschreibung der Implementierung

Alle erforderlich Änderungen werden in der Datei *mip-reg.cc* vorgenommen. Diese enthält im Original die Registrierungsroutinen [1] für Mobile IP, die entsprechend abgewandelt werden. Zunächst werden einige Hilfsroutinen definiert. Die Klasse **MIPBSAgent** definiert Routinen der Basisstation bzw. des FA. Die Klasse **MIPMHAgent** definiert Routinen der Mobile. Die wichtigste Routine ist für diese beiden Klassen die Prozedur *recv*. Alle erforderlichen Routinen sollen folgend kurz beschrieben werden, um einen Überblick über die Implementierung zu verschaffen.

5.2.1 Die Funktion sendOutMessagetoMN

In den Zeilen 1 bis 33 ist die Funktion *sendOutMessagetoMN* definiert. Sie ermöglicht das Senden von Steuernachrichten von der Basisstation zur Mobile. Diese Funktion ist in der Klasse MIPBSAgent definiert. In den Zeilen 6-9 wird auf den Paketheader zugegriffen und der Zugriff auf IP-Header, MIP-Header und Common-Header ermöglicht. Aus diesen Headern können die Quell- und Zieladressen ausgelesen bzw. gesetzt werden, die zum Versenden einer Nachricht erforderlich sind. Desweiteren werden weitere Parameter gesetzt, wie z. B. Lifetime, ID-Nummer oder die Adresse es oFA. Mittels des Befehls *send (pkt, 0)* in Zeile 32 wird die Nachricht versendet.

[1] Alle Zeilennummern beziehen sich auf den Quelltext in Anhang B

5.2.2 Die Funktion sendOutControlMessage

Die Funktion *sendOutControlMessage* funktioniert grundsätzlich genauso wie die Funktion sendOutMessagetoMN. Sie wird in beiden Klassen MIPBSAgent und MIPMHAgent definiert. Diese Nachricht wird dazu verwendet, um Steuernachrichten zwischen Basisationen bzw. Steuernachrichten von der Mobile zur Basisstation zu senden. Beide Prozeduren sind in den Zeilen 35-57 (MIPBSAgent) und 141-162 (MIPMHAgent) definiert.

5.2.3 Die Funktion recv der Klasse MIPBSAgent

In der Funktion *recv* (Zeilen 58-140) der Klasse MIPBSAgent wird abgefragt, ob der FA Nachrichten empfangen hat und wie diese behandelt werden. An dieser Stelle tauchen die Bezeichnungen aus Abschnitt 5.1 wieder auf. Hier wird der ereignisorientierte Charakter von ns-2 deutlich. In der Anweisung *switch->miph-type_* (Zeile 70) wird abgefragt, welche Nachricht gerade abgefragt und wie diese behandelt wird. Zum Weiterleiten der Nachrichten aus Abschnitt 5.1 werden die beiden Hilfsroutinen sendOutControlMessage und sendOutMessagetoMN verwendet. Parameter werden entsprechend gesetzt.

5.2.4 Die Funktion recv der Klasse MIPMHAgent

Die Funktion *recv* (Zeilen 163-255) der Klasse MIPMHAgent fragt ab, ob der MH Nachrichten empfangen hat und wie diese behandelt werden. Hier wird die Funktion sendOutControlMessage zum Weiterleiten der Nachrichten verwendet.

5.3 Das Szenario zur Bewertung der Protokolle

Die Bewertung der Protokolle erfolgt im Netzwerksimulator ns-2 (Version 2.29). Es wird eine hierarchische Anordnung von 16 Basisstationen verwendet, die als Foreign Agent (FA) fungieren. In der Sendereichweite jeder Station befinden sich 10 Mobile. Somit befinden sich insgesamt 160 Mobile in dem Netzwerk. 60 dieser Mobile erzeugen Verkehr, alle anderen befinden sich im Idle Modus. Die aktiven Mobile kommunizieren mit insgesamt 6 Correspondent Hosts, die als CH1 bis CH6 bezeichnet sind. Der Abstand zwischen zwei Basisstationen beträgt 140 m, so dass sich der Sendebereich benachbarter Stationen überlappt. Eine Mobile verliert die Verbindung, wenn diese eine neue Advertisement Nachricht vom neuen Foreign Agent erhält. Der Delay auf den

verdrahteten Links beträgt 2 bzw. 5 ms mit einer Kapazität von 100 Mbit/s. Der Verkehr einer spezifischen Mobile, hier als MH(0) bezeichnet, wird überwacht. Die Mobile MH(0) bewegt sich von der ersten bis zur letzten Basisstation mit einer Geschwindigkeit von 40 km/h. Somit führt MH(0) 15 Handoffs durch. UDP-Verkehr wird in Uplink- und Downlink erzeugt. Es wird die Verkehrsart CBR mit einem Sendeintervall von 20 ms bei einer Paketgröße von 500 Byte verwendet.
Für die Bewertung aller Protokolle wird dieselbe Netzwerktopologie verwendet. Bei HAWAII müssen im TCL-Skript zusätzlich HAWAII-Router definiert werden. Um stabile Ergebnisse zu erhalten, wird diese Messung 10 mal wiederholt. In der aktuellen Implementierung von HAWAII in ns-2 funktioniert der Uplink für mehrere Mobile nicht, so dass für diesen Fall keine Messwerte ermittelt werden können. Abbildung 5.2 zeigt das Szenario, welches für die Simulationen verwendet wird.

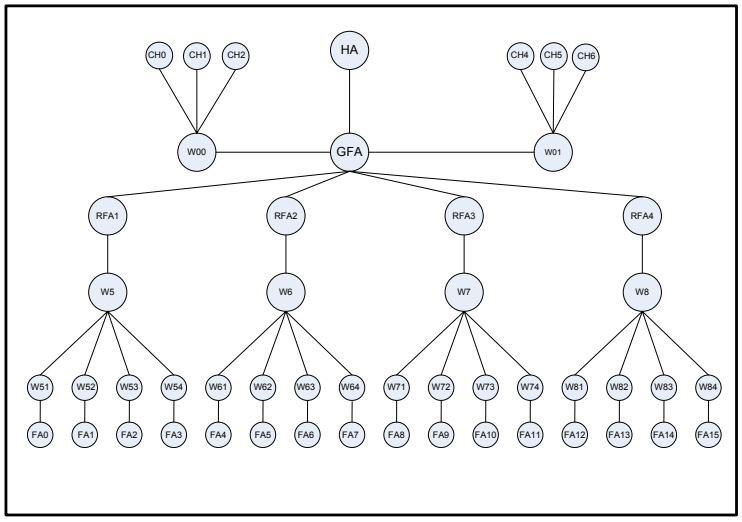

Abbildung 5.2: Szenario für die Simulation

6 Bewertung der untersuchten Protokolle

Zur Bewertung herangezogen werden die vier Protokolle HAWAII, MIP, MIFA und L-MIFA. Untersucht werden die Handoff Latenzzeiten und Paketverlustraten bei UDP-Quellen. Desweiteren soll untersucht werden, wie stark sich die Änderung der Netzlast auf Handoff Latenzzeiten und Paketverlustraten auswirkt. Es sollen Aussagen darüber getroffen werden, welche der Protokolle für Echtzeitanwendungen geeignet sind. Dazu werden sowohl die Durchschnittswerte als auch die Verteilungsfunktionen der Handoff Latenzzeiten und Paketverlustraten dargestellt. Mittels dieser Funktionen sind verlässliche statistische Aussagen über die Protokolle möglich.

6.1 Ermittlung der Handoff Latenzzeiten

Die Handoff Latenzzeit ist bei MIP, HAWAII und MIFA im Uplink die Zeitdauer zwischen dem Aussenden der Registration Request Nachricht und dem Empfang der Registration Reply Nachricht. Bei MIFA im Downlink muss zu dieser Zeit noch die Zeitdauer zwischen dem Aussenden der PFA_Not Nachricht an den alten FA durch den aktuellen FA und den Empfang der PFA_Ack Nachricht durch den aktuellen FA hinzugerechnet werden.
Bei L-MIFA ist die Handoff Latenzzeit die Zeitdauer zwischen dem Aussenden der 2. Registration Reply Nachricht durch den neuen FA und dem Empfang dieser Nachricht durch die Mobile.
Die Netzwerklast wird bei dieser Anordnung durch zufällige Änderung des Sendeintervalls von 40 bis 90 ms der 60 aktiven Mobile variiert.

6.1.1 Handoff Latenzzeiten bei Linkdelay 2 ms

Abbildung 6.1 zeigt die Verteilungsfunktionen der Handoff Latenzzeiten der Protokolle HAWAII, MIP, MIFA in Up- und Downlink sowie L-MIFA bei einem Linkdelay von

2 ms.

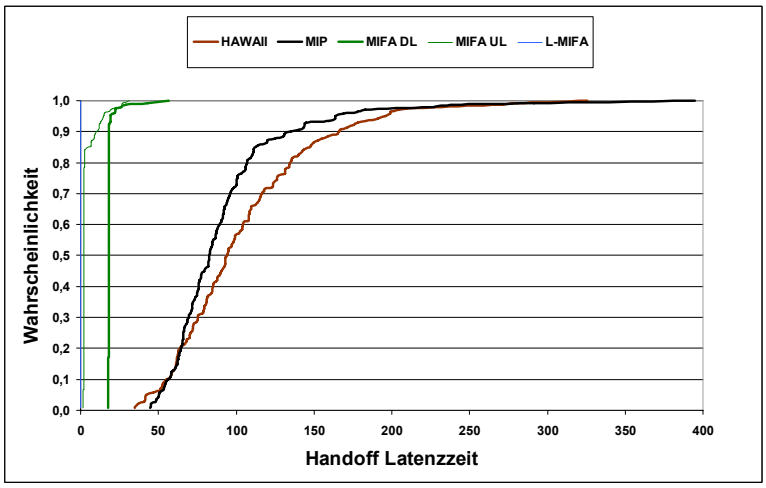

Abbildung 6.1: Verteilung der Handoff Latenzzeiten, Linkdelay 2 ms

MIFA erzielt gegenüber HAWAII und MIP deutlich kürzere Latenzzeiten. Die Verteilungsfunktion von MIFA ist stabiler als bei den anderen Protokollen. Im Downlink von MIFA dauern 95 % aller Handoffs weniger als 19 ms. Im Uplink von MIFA dauern 85 % aller Handoffs weniger als 4 ms. Bei HAWAII und MIP fällt die Handoff Latenzzeit wesentlich höher aus. 90 % aller Handoff bei MIP betragen weniger als 135 ms, bei HAWAII sind es 166 ms. Die Verteilungsfunktion für L-MIFA ist 0. MIFA ist sowohl für hohe als auch niedrige Lasten besser als HAWAII und MIP. HAWAII ist für eine niedrige Netzlast (20 % aller Handoffs) besser als MIP, ab 97 % aller Handoffs sind HAWAII und MIP wieder vergleichbar. Zwischen diesen beiden Schwellwerten ist MIP etwas besser als HAWAII. L-MIFA ist mit einer Latenzzeit von 0 das beste Protokoll. Abbildung 6.2 zeigt die Durchschnittswerte der Handoff Latenzzeiten der untersuchten Protokolle. MIP ist um 10 % besser als HAWAII, MIFA im Downlink ist um 80 % besser als MIP. MIFA im Uplink ist um 79 % besser als MIFA im Downlink. Für L-MIFA ergibt sich ein Durchschnittswert von 0.

6.1.2 Handoff Latenzzeiten bei Linkdelay 5 ms

Abbildung 6.3 zeigt die Verteilungsfunktionen der Handoff Latenzzeiten der Protokolle HAWAII, MIP, MIFA in Up- und Downlink sowie L-MIFA bei einem Linkdelay von

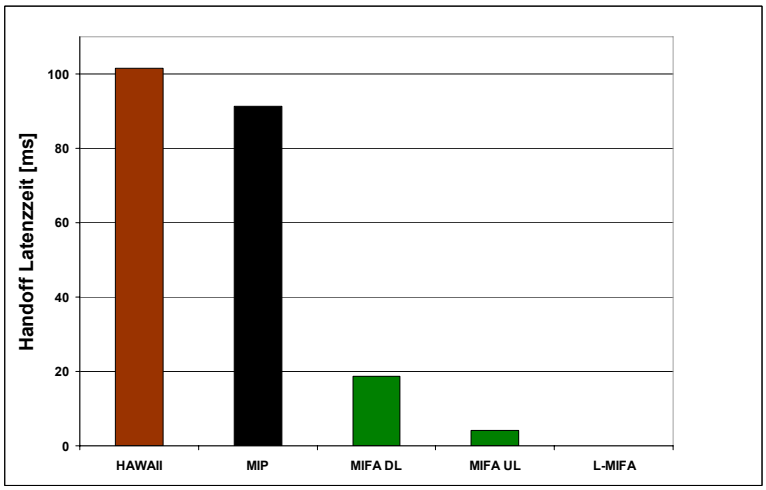

Abbildung 6.2: Durchschnittswerte der Handoff Latenzzeiten, Linkdelay 2 ms

5 ms.
Auch hier unterbietet MIFA in Up- und Downlink HAWAII und MIP deutlich. Für MIFA ist die Verteilungsfunktion stabil, welche bei HAWAII und MIP noch nicht ausreichend stabilisiert ist. Die Handoff Latenzzeit für MIFA im Uplink beträgt für 90 % aller Handoffs nicht mehr als 4 ms, im Downlink sind es 82 ms. Für L-MIFA ergibt sich der Wert 0 für die Verteilungsfunktion und somit die geringsten Werte aller untersuchten Protokolle. Die Handoff Latenzzeiten bei MIP und HAWAII sind wesentlich höher. Die Zeitdauer bei 90 % aller Handoffs beträgt bei HAWAII weniger als 205 ms, bei MIP 160 ms. MIFA erzielt sowohl für niedrige als auch hohe Netzlasten kürzere Latenzzeiten als MIP und HAWAII. HAWAII ist bis 62 % aller Handoffs besser als MIP. Ab 97 % aller Handoffs sind MIP und HAWAII wieder vergleichbar.
Abbildung 6.4 zeigt die Durchschnittswerte der Handoff Latenzzeiten der untersuchten Protokolle. HAWAII ist 0,8 % besser als MIP. MIFA im Downlink ist um 64 % besser als MIP. MIFA im Uplink ist um 92 % besser als MIFA im Downlink. Bei L-MIFA ergibt sich eine durchschnittliche Latenzzeit von 0, was den geringsten Wert darstellt.

6.2 Ermittlung der Paketverlustraten

Analog zu den Betrachtungen für die Handoff Latenzzeiten sollen jetzt die zugehörigen Paketverlustraten je Handoff ermittelt werden.

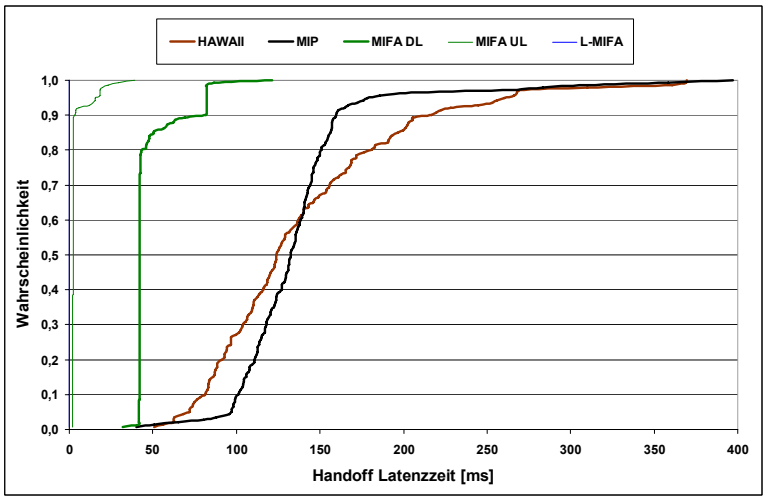

Abbildung 6.3: Verteilung der Handoff Latenzzeiten, Linkdelay 5 ms

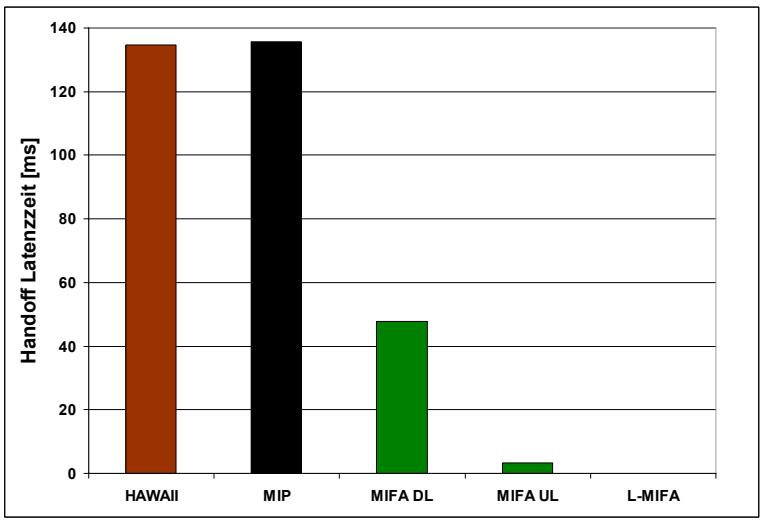

Abbildung 6.4: Durchschnittswerte der Handoff Latenzzeiten, Linkdelay 5 ms

6.2.1 Paketverlustraten bei Linkdelay 2 ms

Abbildung 6.5 zeigt die Verteilungsfunktion der Paketverlustraten bei einem Linkdelay von 2 ms.

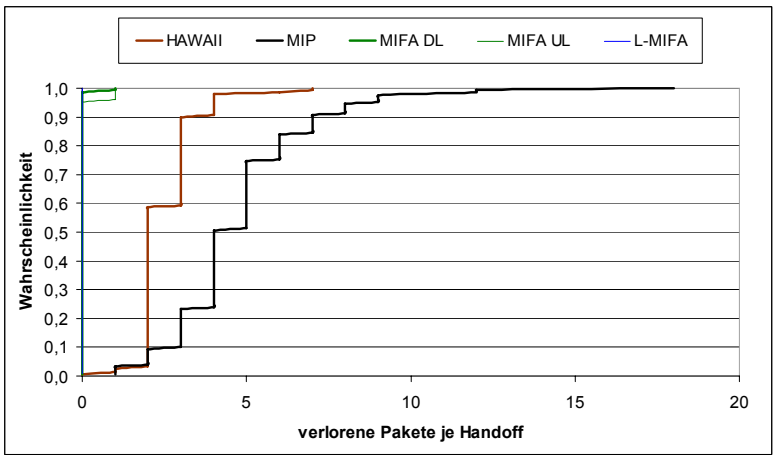

Abbildung 6.5: Verteilung der Paketverlustraten, Linkdelay 2 ms

Bei MIFA treten bis 98 % aller Handoffs überhaupt keine Paketverluste auf. HAWAII und MIP sind deutlich schlechter als MIFA, wobei HAWAII deutlich besser als MIP ist. Bei HAWAII werden bis 90 % aller Handoffs nicht mehr als drei Pakete je Handoff gedroppt, bei MIP sind es sieben Pakete. Bei L-MIFA kommt es zu keinen Paketverlusten. MIFA ist für geringe und hohe Netzlasten besser als HAWAII und MIP. HAWAII ist ebenfalls für geringe und hohe Netzlasten besser als MIP. Die Paketverluste für MIFA im Up- und Downlink sind vergleichbar. L-MIFA erzielt die besten Werte, da keine Pakete gedroppt werden.
Abbildung 6.6 zeigt die Durchschnittswerte der Paketverlustraten bei einem Linkdelay von 2 ms. HAWAII ist um 47 % besser als MIP. MIFA im Downlink ist um 95 % besser als HAWAII. MIFA im Uplink ist um 75 % schlechter als MIFA im Downlink. L-MIFA schneidet am besten ab, da keine Paketverluste auftreten.

6.2.2 Paketverlustraten bei Linkdelay 5 ms

Abbildung 6.7 zeigt die Verteilungsfunktion der Paketverlustraten bei einem Linkdelay von 5 ms.

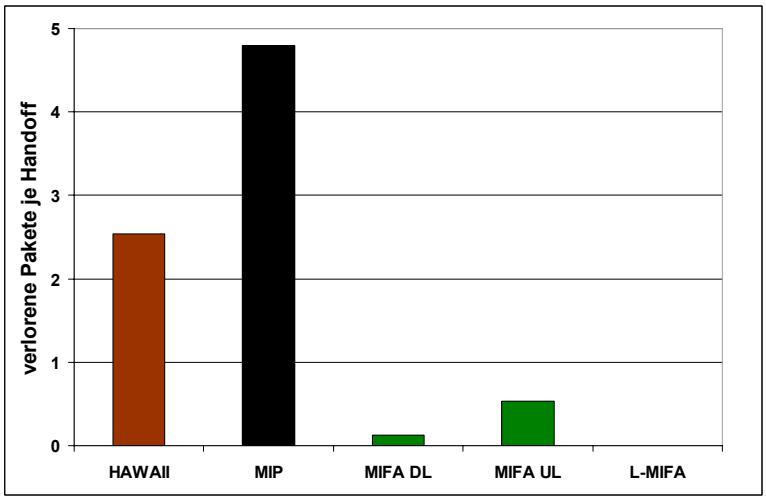

Abbildung 6.6: Durchschnittswerte der Paketverlustraten, Linkdelay 2 ms

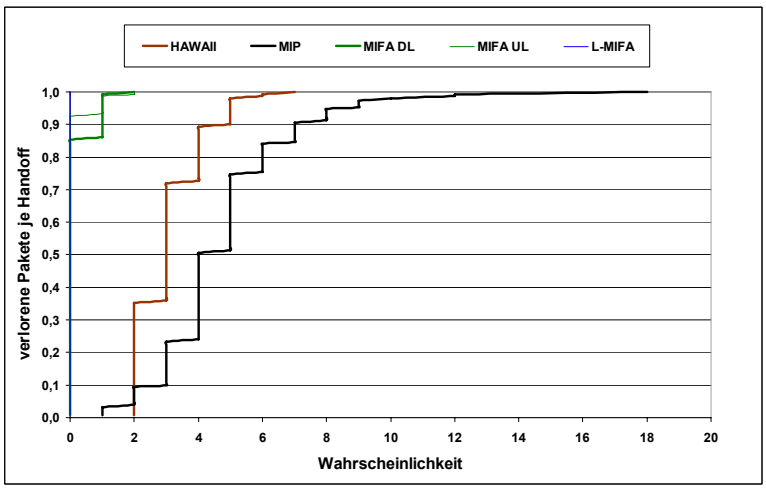

Abbildung 6.7: Verteilung der Paketverlustraten, Linkdelay 5 ms

MIFA weist deutlich geringere Paketverlustraten gegenüber HAWAII und MIP auf, bis 85 % aller Handoffs im Downlink und 95 % aller Handoffs im Uplink treten keine Paketverluste auf. Die Paketverlustraten von HAWAII und MIP fallen deutlich höher aus. Die Paketverlustrate bei HAWAII beträgt nicht mehr als 4 Pakete je Handoff bei 90 % aller Handoffs, bei MIP sind es 7 Pakete. MIFA ist sowohl für geringe als auch hohe Netzlasten besser als HAWAII und MIP, HAWAII ist besser als MIP. L-MIFA ist das beste Protokoll, da keine Paketverluste auftreten.
Abbildung 6.8 zeigt die Durchschnittswerte der Paketverlustraten bei einem Linkdelay von 5 ms. Bezugnehmend auf die Durchschnittswerte ist HAWAII um 59 % besser als MIP. MIFA im Downlink ist um 50 % besser als HAWAII. MIFA im Uplink ist um 43 % besser als MIFA im Downlink. Bei L-MIFA kommt es zu keinen Paketverlusten.

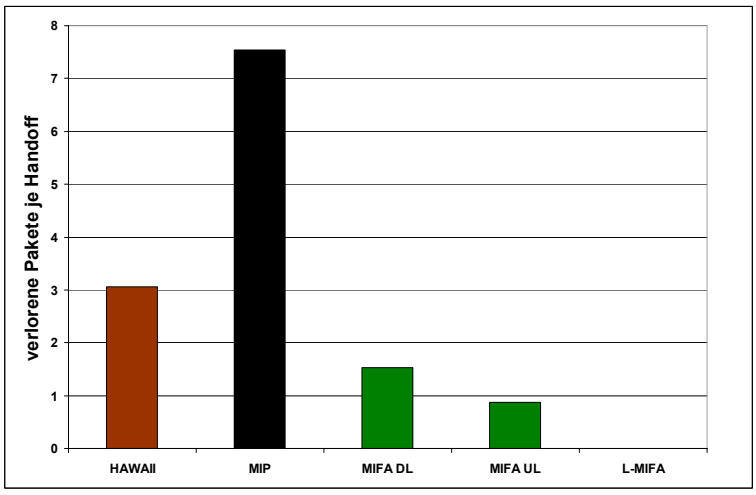

Abbildung 6.8: Durchschnittswerte der Paketverlustraten, Linkdelay 5 ms

6.3 Einfluss der Netzlast

In diesem Abschnitt soll untersucht werden, wie stark sich die Änderung der Netzlast auf die Handoff Latenzzeiten und Paketverlustraten auswirken. Es werden die gleichen Szenarioeinstellungem wie im letzten Abschnitt verwendet. In diesem Szenario wird die Anzahl aktiver Mobile verändert, schrittweise werden eine, zehn, 30 und 60 aktive Mobile im Netz mit Verkehr geschaltet. Die restlichen Mobile verbleiben im Idle Modus. Der Delay-Wert der verdrahteten Links wird auf 5 ms gesetzt. Hierzu werden

für die Protokolle HAWAII, MIP, MIFA im Up- und Downlink die statistischen Verteilungsfunktionen für Handoff Latenzzeiten und Paketverlustraten grafisch dargestellt. Desweiteren werden die Durchschnittswerte für alle Protokolle verglichen. Innerhalb eines der vier genannten Szenarien bleibt die Netzlast konstant.

6.3.1 Einfluss der Netzlast auf HAWAII

Handoff Latenzzeiten Abbildung 6.9 zeigt die statistische Verteilungsfunktion der Handoff Latenzzeiten des HAWAII-Protokolls.

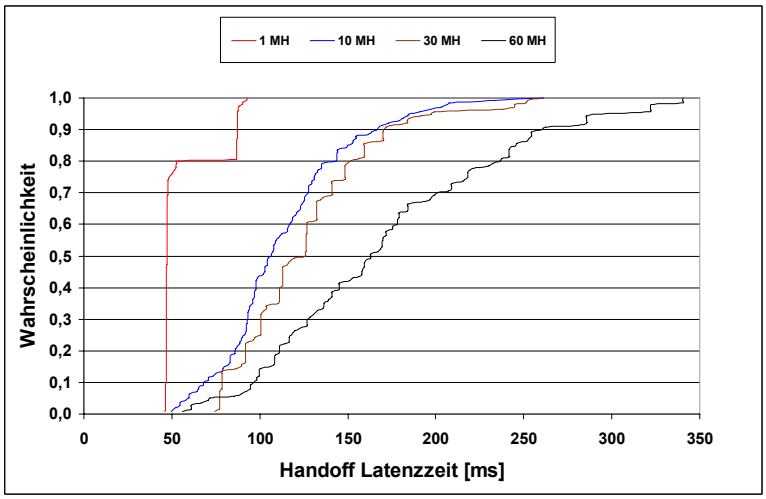

Abbildung 6.9: Verteilung der Handoff Latenzeiten bei Laständerung - HAWAII

Für eine Mobile beträgt die Handoff Latenzzeit für 90 % aller Handoffs nicht mehr als 87 ms. Für 10 Mobile beträgt dieser Wert 167 ms, für 30 Mobile sind es 171 ms. Für 60 Mobile wird ein Spitzenwert von 254 ms erreicht. Die Verteilungsfunktionen sind stabil. Die Handoff Latenzzeiten der Verteilungsfunktion für eine Mobile sind deutlich besser als die für 10 Mobile. Die Werte für 10 Mobile sind geringfügig besser als die für 30 Mobile, die Verteilungsfunktion für 60 Mobile fällt deutlich am schlechtesten aus.

Paketverlustraten In Abbildung 6.10 sind die Verteilungsfunktionen der Paketverlustraten angegeben. Die Höchstwerte für die Paketverluste je Handoff betragen für 90 % aller Handoffs ein Paket für eine Mobile, drei Pakete bei 10 und 30 Mobilen und 5 Pakete bei 60 Mobilen. Die Werte der Verteilungsfunktion für eine Mobile sind am

besten. Die Werte für 10 und 30 Mobile liegen relativ nahe zusammen. Die Verteilungsfunktion für 60 Mobile ist am schlechtestens.

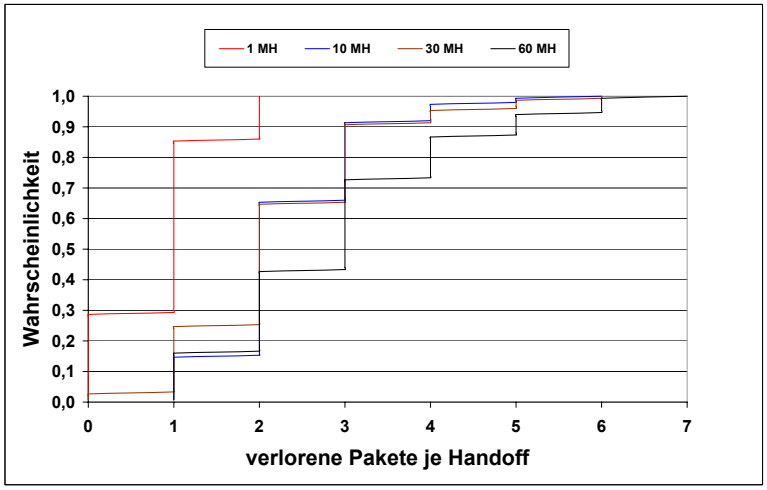

Abbildung 6.10: Verteilung der Paketverlustraten bei Laständerung - HAWAII

Aufgrund der obigen Diskussion ist zu erkennen, dass durch die Erhöhung der Netzlast die Latenzzeiten und Paketverlustraten sich deutlich verschlechtern. HAWAII wird von einer Änderung der Netzlast stark beeinflusst.

6.3.2 Einfluss der Netzlast auf MIP

Handoff Latenzzeiten Abbildung 6.11 stellt die Verteilung der Handoff Latenzzeiten bei MIP dar. Für 90 % aller Handoffs betragen die Latenzzeiten nicht mehr als 98 ms bei einer Mobile, 132 ms bei 10 Mobilen, 177 ms bei 30 Mobilen und 203 ms bei 60 Mobilen.
Die Verteilungsfunktion für eine Mobile ist die beste und bleibt bis 90 % aller Handoffs relativ konstant. Die Funktionen für 10 und 30 Mobile sind schlechter als die für eine Mobile und bleiben für niedrige Netzlasten bis ca. 40 % aller Handoffs fast gleich. Ab 40 % ist die Verteilungsfunktion für 30 Mobile schlechter als die für 10 Mobile und verschlechtert sich somit für hohe Netzlasten. Die Werte der Funktion für 60 Mobile sind am höchsten.

Paketverlustraten Abbildung 6.12 zeigt die Verteilungsfunktionen der Paketverlustraten für MIP bei Laständerung. Bei einer aktiven Mobile werden bis zu 90 % aller

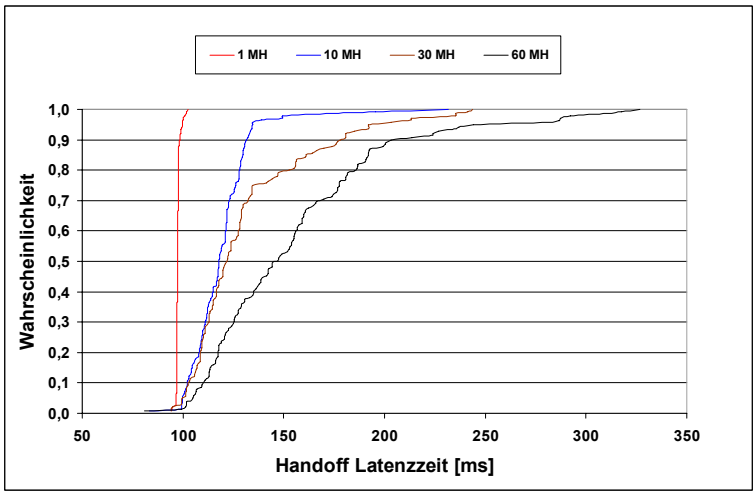

Abbildung 6.11: Verteilung der Handoff Latenzzeiten bei Laständerung - MIP

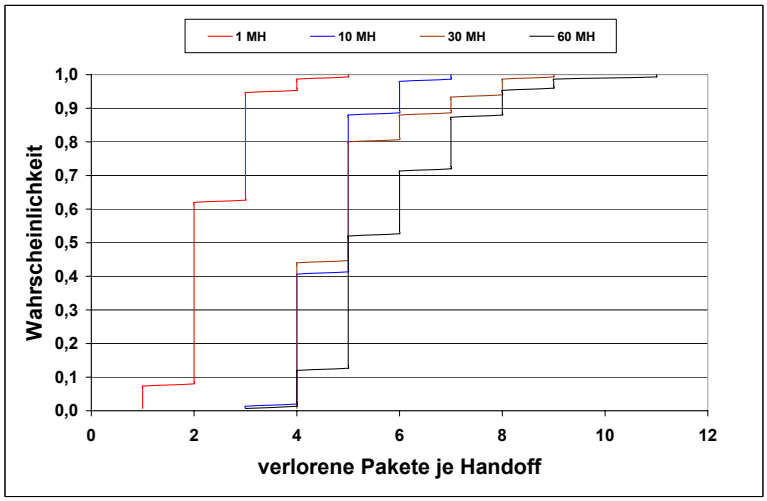

Abbildung 6.12: Verteilung der Paketverlustraten bei Laständerung - MIP

Handoffs nicht mehr als drei Pakete gedroppt, bei 10 Mobilen sind es sechs Pakete, bei 30 Mobilen sieben Pakete und bei 60 Mobilen beträgt dieser Wert acht gedroppte Pakete je Handoff. Die Verteilung der Paketverluste für eine Mobile ist am geringsten für niedrige und hohe Netzlasten. Die Verteilung für 10 und 30 Mobile sind in etwa gleich, aber schlechter als die Verteilung für eine Mobile. Die Verteilungsfunktion für 60 Mobile ist für niedrige und hohe Netzlasten am schlechtesten.
Aus den obigen Messungen ist klar zu erkennen, dass die Erhöhung der Netzlast einen Anstieg der Handoff Latenzzeiten und Paketverlustraten zur Folge hat. Mobile IP ist lastabhängig.

6.3.3 Einfluss der Netzlast auf MIFA

Handoff Latenzzeiten Abbildung 6.13 zeigt die Verteilung der Handoff Latenzzeiten bei MIFA im Downlink.

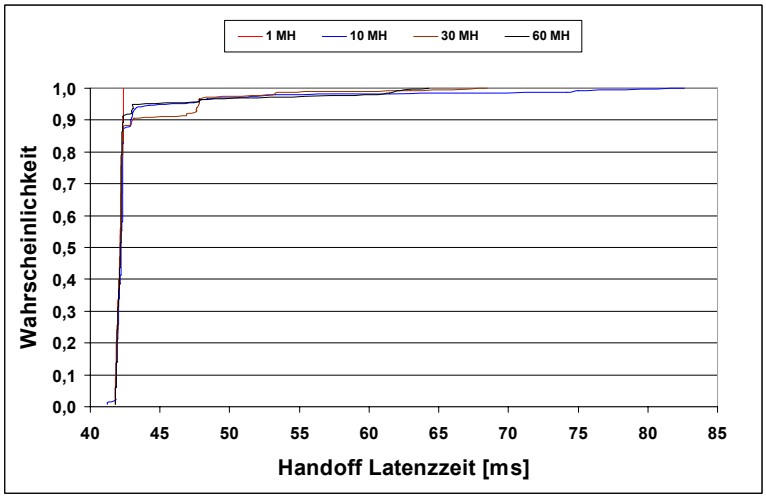

Abbildung 6.13: Handoff Latenzzeiten bei Laständerung - MIFA Downlink

Bis auf geringe Abweichungen ergeben sich ähnliche Werte für eine, 10, 30 und 60 Mobile von weniger als 42 ms bei 85 % aller Handoffs. Bis auf wenige abweichende Werte ergeben sich somit fast gleichbleibende stabile Verteilungsfunktionen trotz veränderter Netzlast. Diese gleich bleibenden Werte lassen sich durch die Arbeitsweise des MIFA-Protokolls erklären. Ein von der Mobile ausgesendete Registration Request Nachricht wird direkt durch den aktuellen FA mit einer Registration Reply Nachricht

beantwortet. Die Zeitdauer dieser Nachricht wird durch die Netzlast nur gering beeinflusst. Hinzu kommt im Downlink die Zeitdauer des Tunnel über den Previous Foreign Agent, die ebenfalls wenig von der Netzlast beeinflusst wird.
In Abbildung 6.14 ist die Verteilung der Handoff Latenzzeiten für eine, 10, 30 und 60 aktive Mobile für MIFA im Uplink dargestellt.

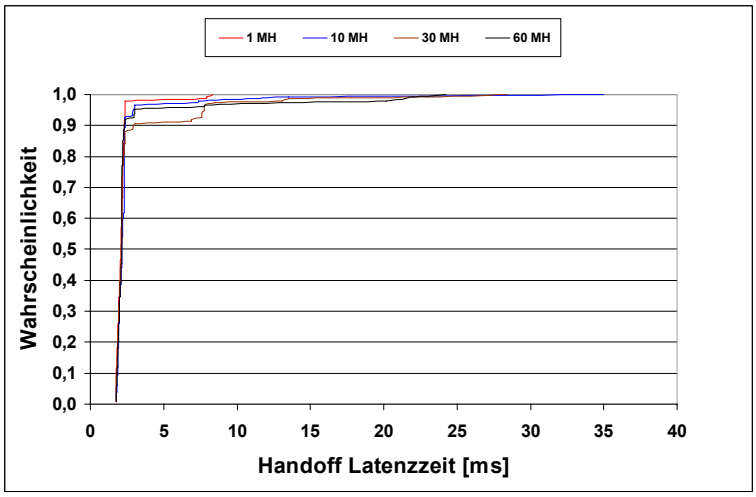

Abbildung 6.14: Handoff Latenzzeiten bei Laständerung - MIFA Uplink

Auch im Uplink ergeben sich vergleichbare Werte für alle vier Verteilungsfunktionen. Die Dauer der Handoffs beträgt für alle 4 Funktionen nicht mehr als 2 ms bei 88 % aller Handoffs. Die Verteilungsfunktionen sind stabil und ändern sich durch die Veränderung der Netzlast kaum. Wie im Downlink können diese Werte durch die Funktionsweise des MIFA-Protokolls erklärt haben. Die Zeitdauer der Initialregistrierung (Registration Request - Registration Reply) bleibt relativ konstant. In Up- und Downlink bei MIFA entstehen hohe Latenzzeiten erst ab sehr hohen Netzlasten über 90 %.

Paketverlustraten In Abbildung 6.15 sind die Paketverlustraten für MIFA im Downlink bei Laständerung dargestellt.
Bei 97 % aller Handoffs treten in allen Szenarien keine Paketverluste im Downlink auf. Mit einer Mobile treten überhaupt keine Paketverluste auf, bei 10 Mobilen ist es maximal ein Paket je Handoff. Bei 30 Mobilen werden maximal zwei Pakete je Handoff gedroppt, bei 60 aktiven Mobilen sind drei Pakete. Die Paketverteilungsfunktionen unterscheiden sich nur wenig. Die Änderung der Netzlast wirkt sich nur sehr gering

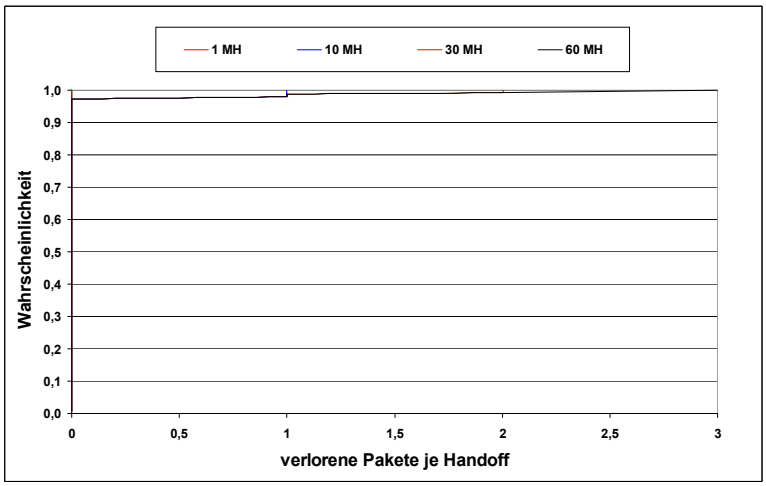

Abbildung 6.15: Paketverlustraten bei Laständerung - MIFA Downlink

auf die Verteilung der Paketverlustraten aus, da die Zeitdauer der Initialregistrierung und der Tunnelzeit fast konstant bleiben. Die Verteilungsfunktionen sind stabil, um statistisch aussagekräftig zu sein.
In Abbildung 6.16 sind die Paketverlustraten für MIFA im Uplink bei Laständerung dargestellt. Bei allen Szenarien werden bei 95 % aller Handoffs keine Pakete gedroppt. Wie im Downlink unterscheiden sich die Verteilungsfunktionen nur wenig, maximal werden zwischen null und drei Pakete gedroppt.
Anhand der gemessenen Werte ist klar zu erkennen, dass eine Änderung der Last sich nur kaum auf die Handoff Latenzzeiten und Paketverlustraten auswirkt. MIFA ist ein lastunabhängiges Protokoll.

6.3.4 Einfluss der Netzlast auf L-MIFA

Bei L-MIFA betragen die Handoff Latenzzeiten und Paketverlustraten in allen Szenarien 0. Wie bei MIFA wirkt sich die Last nicht aus, L-MIFA ist somit lastunabhängig.

6.3.5 Vergleich von HAWAII, MIP, MIFA und L-MIFA

Handoff Latenzzeiten Abbildung 6.17 zeigt die Durchschnittswerte der Handoff Latenzzeiten der vier Protokolle HAWAII, MIP, MIFA und L-MIFA mit Laständerung. Der Einfluss der Last soll hier bewertet werden. Die Handoff Latenzzeit von L-MIFA

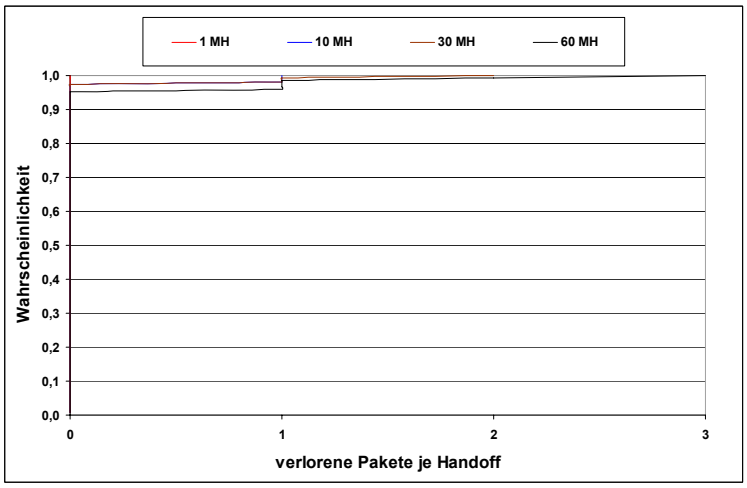

Abbildung 6.16: Paketverlustraten bei Laständerung - MIFA Uplink

beträgt in allen 4 Szenarien 0.

Vergleich für eine Mobile Für eine Mobile im Netz ist HAWAII um 37 % besser als MIP. MIFA im Downlink ist um 24 % besser als HAWAII. MIFA im Uplink ist um 95 % als MIFA im Downlink.

Vergleich für zehn Mobile Bei zehn Mobilen im Netz erzielt HAWAII eine um 9 % bessere Handoff Latenzzeit als MIP. MIFA im Downlink ist um 61 % besser als HAWAII. MIFA im Uplink ist um 95 % besser als im MIFA im Downlink.

Vergleich für 30 Mobile Hier ist HAWAII um 7 % besser als MIP. MIFA im Downlink ist um 65 % besser als HAWAII. MIFA im Uplink ist um 93 % besser als MIFA im Downlink.

Vergleich für 60 Mobile HAWAII ist in diesem Szenario um 7 % schlechter als MIP. MIFA im Downlink ist um 74 % besser als HAWAII. MIFA im Downlink ist um 93 % besser als MIFA im Uplink.

Paketverlustraten Abbildung 6.18 zeigt die Durchschnittswerte der Paketverlustraten. Der Einfluss der Laständerung auf die Paketverlustraten soll hier untersucht werden. Die Paketverluste bei L-MIFA betragen in allen Szenarien 0.

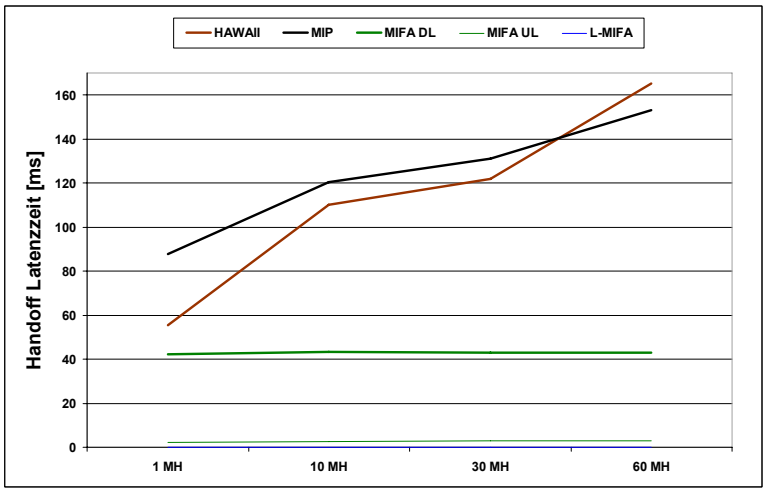

Abbildung 6.17: Durchschnittswerte der Handoff Latenzzeiten bei Laständerung

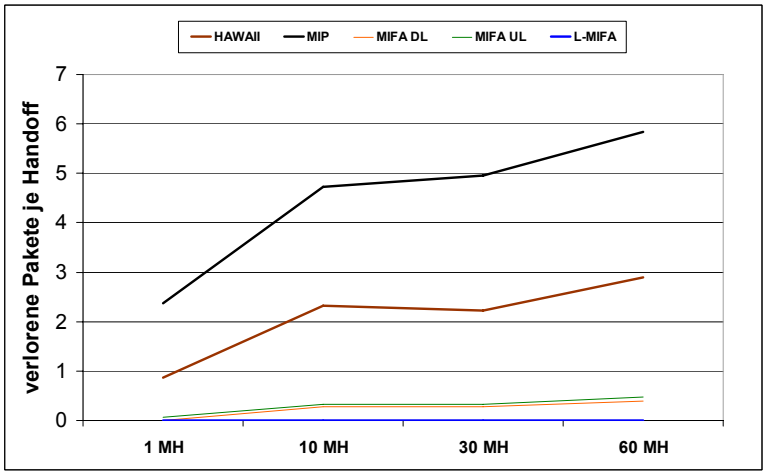

Abbildung 6.18: Durchschnittswerte der Paketverlustraten bei Laständerung

Vergleich für eine Mobile Die Paketverlustraten je Handoff für HAWAII sind hier um 67 % besser als MIP. Bei MIFA im Downlink gibt es keine Paketverluste, MIFA im Uplink ist 92 % besser als HAWAII.

Vergleich für zehn Mobile Hier ist HAWAII nur noch 51 % besser als MIP. MIFA im Downlink ist hier um 88 % und MIFA im Uplink um 86 % besser als HAWAII.

Vergleich für 30 Mobile HAWAII hat um 55 % geringere Paketverluste als MIP. MIFA im Downlik erzielt um 88 %, MIFA im Uplink um 85 % bessere Werte.

Vergleich für 60 Mobile HAWAII ist um 50 % besser als MIP. MIFA im Downlink ist um 86 %, MIFA im Uplink um 84 % besser als HAWAII.
Die Erhöhung der Last wirkt sich stark auf HAWAII und MIP aus, wobei MIP deutlich schlechter abschneidet als HAWAII. HAWAII und MIP sind somit stark lastabhängige Protokolle. Auf MIFA und L-MIFA wirkt sich die Last nur wenig aus. MIFA und L-MIFA sind somit lastunabhängig.

7 Zusammenfassung und Ausblick

7.1 Ergebnisse

Ziel dieser Arbeit war die Implementierung und Evaluierung des verbesserten Mobilitätsprotokolls L-MIFA. L-MIFA enthält Ansätze, um insbesondere die Handoff Latenzzeit bei Wechsel des Netzzugangspunktes stark zu verkürzen und die dabei auftretenden Paketverluste zu verringern.
Im Rahmen dieser Arbeit wurde L-MIFA in den Netzwerksimulator ns-2 implementiert. Anschließend wurden Performanzanalysen der Protokolle MIP, HAWAII, MIFA und L-MIFA zur Ermittlung der Handoff Latenzzeiten und Paketverlustraten durchgeführt. Des weiteren wurde der Einfluss der Last untersucht.
Die Ergebnisse der Analyse haben gezeigt, dass die Handoff Latenzzeiten und Paketverlustraten von MIFA und L-MIFA deutlich deutlich geringer sind als bei HAWAII und MIP. Bei L-MIFA liegen diese Werte nahe Null. Desweiteren hat sich gezeigt, dass sich eine Änderung der Last auf MIFA und L-MIFA kaum auswirkt, wahrend dies bei HAWAII und MIP einen starken Anstieg der Handoff Latenzzeiten und der damit verbundenen Paketverlustraten bewirkt. MIFA und L-MIFA erfüllen mit einer durchschnittlichen Handoff Latenzzeit von unter 50 ms Echtzeitanforderungen. Damit sind diese beide Protokolle für Echtzeitanwednungen wie beispielsweise VoIP oder Videostreaming geeignet.

7.2 Ausblick

In Zukunft sind weitere Performanzanalysen geplant. Es ist zu untersuchen, wie stark sich Handoff Latenzzeiten und Paketverlustraten durch die Änderung der Geschwindigkeit der Mobile ändern. Des weiteren soll der TCP-Durchsatz evaluiert werden. Ebenfalls zu analysieren sind weitere Protokolle wie CIP, HMIP sowie weitere Mobilitätsprotokolle der IPv6-Generation, wie z. B. Fast Hierarchical Mobile IP (FHMIP). Weiterhin ist der Einfluss der Netztopologien zu untersuchen, beispielsweise einer Meshtopologie.

Anhang

A Szenarioeinstellungen

A.1 MIP, MIFA und L-MIFA

Das folgende Tcl-Skript wurde für die Simulationen im Kapitel 6 für die Protokolle MIP, MIFA und L-MIFA verwendet.

```
# MIP, MIFA and L-MIFA simulation in ns-2
# Copyright (c) 2007 Technische Universität Ilmenau.
# All rights reserved.
# first author              : Christian Kellner
# co-author and supervisor : Dipl.-Ing. Ali Diab
set dtime     [lindex $argv 0]
set lostfile [lindex $argv 1]
set timefile [lindex $argv 2]
set sendfile [lindex $argv 3]
puts "delay is $dtime"
puts "lostfile is $lostfile"
puts "timefile is $timefile"
puts "sendfile is $sendfile"
# Increase Queue in link
Queue set limit_ 200
# Define options for wireless scenario
set opt(chan)    Channel/WirelessChannel
set opt(prop)    Propagation/TwoRayGround
set opt(netif)   Phy/WirelessPhy
set opt(mac)     Mac/802_11
set opt(ifq)     CMUPriQueue
set opt(ll)      LL
set opt(ant)     Antenna/OmniAntenna
set opt(ifqlen)          50
set opt(nn)              161
set opt(adhocRouting)    NOAH
set opt(cp)      ""
set opt(sc)      ""
set opt(x)       2200
set opt(y)       2200
set opt(seed)    0.0
set opt(stop)    300.0
set opt(ftp1-start)       0.0
set num_wired_nodes       12
set FaDist                140.0

Agent/TCP set sport_             0
Agent/TCP set dport_             0
Agent/TCP set packetSize_        1460
Queue/DropTail/PriQueue set Prefer_Routing_Protocols 1
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 0.2
Antenna/OmniAntenna set Gr_ 0.2

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
```

```
Phy/WirelessPhy set Rb_ 2*1e6
Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0

# check for boundary parameters and random seed
if { $opt(x) == 0 || $opt(y) == 0 } {
  puts "No X-Y boundary values given for wireless topology\n"
}
if {$opt(seed) > 0} {
  puts "Seeding RandomCOA number generator with $opt(seed)\n"
  ns-random $opt(seed)
}
# Registration interval in Mobile
Agent/MIPMH set reg_rtx_ 1.0
# create simulator instance
set ns_   [new Simulator]
# use scheduler for shorter simulation time
$ns_ use-scheduler Heap
# all nodes in domain of Home Agent
# set up for hierarchical routing
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 22
lappend cluster_num 1 1 1 1 1 1 1 1 1 1 \
 1 1 1 1 1 1 1 1 1 1 1 1
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 10 170 6 6 6 6 1 1 1 \
1 1 1 1 1 1 1 1 1 1 1 1
AddrParams set nodes_num_ $eilastlevel ;# of each domain

set f2 [open $sendfile w]
set f1 [open $timefile w]
set f0 [open $lostfile w]

# suppress huge trace file
set tracefd [open /dev/null w]
## set namtrace [open out.nam w]
set namtrace [open /dev/null w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)
# tell nam the stoptime
$ns_ nam-end-wireless "$opt(stop)"
# Create topography object
set topo    [new Topography]
# define topology
$topo load_flatgrid $opt(x) $opt(y)

# create God 1 for HA and  1 for every FA
create-god [expr 17 + $opt(nn)]

#create wired nodes for domain 0
set W0  [$ns_ node 0.0.0]
set W01 [$ns_ node 0.0.1]
set W00 [$ns_ node 0.0.2]
set CH1 [$ns_ node 0.0.3]
set CH2 [$ns_ node 0.0.4]
set CH3 [$ns_ node 0.0.5]
set CH4 [$ns_ node 0.0.6]
set CH5 [$ns_ node 0.0.7]
set CH6 [$ns_ node 0.0.8]
#create wired nodes for RFA domains
set W1  [$ns_ node 2.0.0]
set W5  [$ns_ node 2.0.1]
set W51 [$ns_ node 2.0.2]
set W52 [$ns_ node 2.0.3]
set W53 [$ns_ node 2.0.4]
set W54 [$ns_ node 2.0.5]
set W2  [$ns_ node 3.0.0]
set W6  [$ns_ node 3.0.1]
set W61 [$ns_ node 3.0.2]
set W62 [$ns_ node 3.0.3]
set W63 [$ns_ node 3.0.4]
set W64 [$ns_ node 3.0.5]
```

```
126 set W3  [$ns_ node 4.0.0]
    set W7  [$ns_ node 4.0.1]
    set W71 [$ns_ node 4.0.2]
    set W72 [$ns_ node 4.0.3]
    set W73 [$ns_ node 4.0.4]
131 set W74 [$ns_ node 4.0.5]
    set W4  [$ns_ node 5.0.0]
    set W8  [$ns_ node 5.0.1]
    set W81 [$ns_ node 5.0.2]
    set W82 [$ns_ node 5.0.3]
136 set W83 [$ns_ node 5.0.4]
    set W84 [$ns_ node 5.0.5]

    # Configure for ForeignAgent and HomeAgent nodes
    $ns_ node-config -mobileIP ON \
141                  -adhocRouting $opt(adhocRouting) \
                     -llType $opt(ll) \
                     -macType $opt(mac) \
                     -ifqType $opt(ifq) \
                     -ifqLen $opt(ifqlen) \
146                  -antType $opt(ant) \
                     -propType $opt(prop) \
                     -phyType $opt(netif) \
                     -channelType $opt(chan) \
                         -topoInstance $topo \
151                  -wiredRouting ON \
                         -agentTrace OFF \
                     -routerTrace OFF \
                     -macTrace OFF
    # Create HA ...
156 set HA [$ns_ node 1.0.0]
    # Create sixteen FAs
    for {set b 0} {$b < 16} {incr b} {
    set FA($b) [$ns_ node [expr 6 +$b].0.0]
    $ns_ at 0.0 "$FA($b)  label FA($b)"
161 $FA($b) color "red"
    }
    $W0  color "maroon"
    $CH1 color "purple"
    $CH2 color "purple"
166 $CH3 color "purple"
    $CH4 color "purple"
    $CH5 color "purple"
    $CH6 color "purple"
    $HA  color "red"
171 $ns_ at 0.0 "$W0  label GFA"
    $ns_ at 0.0 "$W00  label W00"
    $ns_ at 0.0 "$W01  label W01"
    $ns_ at 0.0 "$W01  label W01"
    $ns_ at 0.0 "$CH1  label CH1"
176 $ns_ at 0.0 "$CH2  label CH2"
    $ns_ at 0.0 "$CH3  label CH3"
    $ns_ at 0.0 "$CH4  label CH4"
    $ns_ at 0.0 "$CH5  label CH5"
    $ns_ at 0.0 "$CH6  label CH6"
181 $ns_ at 0.0 "$HA   label HA"
    $ns_ at 0.0 "$W1  label W1"
    $ns_ at 0.0 "$W2  label W2"
    $ns_ at 0.0 "$W3  label W3"
    $ns_ at 0.0 "$W4  label W4"
186 $ns_ at 0.0 "$W5  label W5"
    $ns_ at 0.0 "$W51  label W51"
    $ns_ at 0.0 "$W52  label W52"
    $ns_ at 0.0 "$W53  label W53"
    $ns_ at 0.0 "$W54  label W54"
191 $ns_ at 0.0 "$W6   label W6"
    $ns_ at 0.0 "$W61  label W61"
    $ns_ at 0.0 "$W62  label W62"
    $ns_ at 0.0 "$W63  label W63"
    $ns_ at 0.0 "$W64  label W64"
196 $ns_ at 0.0 "$W7   label W7"
    $ns_ at 0.0 "$W71  label W71"
    $ns_ at 0.0 "$W72  label W72"
    $ns_ at 0.0 "$W73  label W73"
```

```
    $ns_ at 0.0 "$W74  label W74"
201 $ns_ at 0.0 "$W8   label W8"
    $ns_ at 0.0 "$W81  label W81"
    $ns_ at 0.0 "$W82  label W82"
    $ns_ at 0.0 "$W83  label W83"
    $ns_ at 0.0 "$W84  label W84"
206
    # Position (fixed) for base-station nodes (HA & FA).
    $HA set X_ 1000.000000000000
    $HA set Y_ 10.000000000000
    $HA set Z_ 0.000000000000
211
    for {set k 0} {$k < 16} {incr k} {
    $FA($k) set X_ [expr $FaDist * $k]
    $FA($k) set Y_ [expr $FaDist * $k]
    $FA($k) set Z_ 0.0
216 }
    # create a mobilenode that would be moving between HA and FA.
    # note address of MH indicates its in the same domain as HA.
    $ns_ node-config -wiredRouting OFF
    for {set s 0} {$s < $opt(nn)} {incr s} {
221 set MH($s) [$ns_ node 1.0.[expr $s + 1]]
    set node_($s) $MH($s)
    set HAaddress [AddrParams addr2id [$HA node-addr]]
    [$MH($s) set regagent_] set home_agent_ $HAaddress
    $ns_ at 0.0 "$MH($s)  label MH($s)"
226 }
    ## observed MH green; others blue
    $MH(0) color "green"
    for {set s 1} {$s < $opt(nn)} {incr s} {
    $MH($s) color "blue"
231 }
    $MH(0) set X_ 12.0
    $MH(0) set Y_ 12.0

    ## ten nodes at BS1
236 for {set i 1} {$i < 11} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 10 60]
    set ypos [$rng uniform 10 60]
    $MH($i) set X_ $xpos
241 $MH($i) set Y_ $ypos
    }
    ## ten nodes at BS2
    for {set i 11} {$i < 21} {incr i} {
    set rng [new RNG]
246 set xpos [$rng uniform 80 200]
    set ypos [$rng uniform 80 200]
    $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
    }
251 ## ten nodes at BS3
    for {set i 21} {$i < 31} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 220 340]
    set ypos [$rng uniform 220 340]
256 $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
    }
    ## ten nodes at BS4
    for {set i 31} {$i < 41} {incr i} {
261 set rng [new RNG]
    set xpos [$rng uniform 360 480]
    set ypos [$rng uniform 360 480]
    $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
266 }
    ## ten nodes at BS5
    for {set i 41} {$i < 51} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 500 620]
271 set ypos [$rng uniform 500 620]
    $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
```

```
    }
    ## ten nodes at BS6
276 for {set i 51} {$i < 61} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 640 760]
    set ypos [$rng uniform 640 760]
    $MH($i) set X_ $xpos
281 $MH($i) set Y_ $ypos
    }
    ## ten nodes at BS7
    for {set i 61} {$i < 71} {incr i} {
    set rng [new RNG]
286 set xpos [$rng uniform 780 900]
    set ypos [$rng uniform 780 900]
    $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
    }
291 ## ten nodes at BS8
    for {set i 71} {$i < 81} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 920 1040]
    set ypos [$rng uniform 920 1040]
296 $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
    }
    ## ten nodes at BS9
    for {set i 81} {$i < 91} {incr i} {
301 set rng [new RNG]
    set xpos [$rng uniform 1060 1180]
    set ypos [$rng uniform 1060 1180]
    $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
306 }
    ## ten nodes at BS10
    for {set i 91} {$i < 101} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 1200 1320]
311 set ypos [$rng uniform 1200 1320]
    $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
    }
    ## ten nodes at BS11
316 for {set i 101} {$i < 111} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 1340 1460]
    set ypos [$rng uniform 1340 1460]
    $MH($i) set X_ $xpos
321 $MH($i) set Y_ $ypos
    }
    ## ten nodes at BS12
    for {set i 111} {$i < 121} {incr i} {
    set rng [new RNG]
326 set xpos [$rng uniform 1480 1600]
    set ypos [$rng uniform 1480 1600]
    $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
    }
331 ## ten nodes at BS13
    for {set i 121} {$i < 131} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 1620 1740]
    set ypos [$rng uniform 1620 1740]
336 $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
    }
    ## ten nodes at BS14
    for {set i 131} {$i < 141} {incr i} {
341 set rng [new RNG]
    set xpos [$rng uniform 1760 1880]
    set ypos [$rng uniform 1760 1880]
    $MH($i) set X_ $xpos
    $MH($i) set Y_ $ypos
346 }
    ## ten nodes at BS15
```

```
for {set i 141} {$i < 151} {incr i} {
set rng [new RNG]
set xpos [$rng uniform 1900 2020]
set ypos [$rng uniform 1900 2020]
$MH($i) set X_ $xpos
$MH($i) set Y_ $ypos
}
## ten nodes at BS16
for {set i 151} {$i < 161} {incr i} {
set rng [new RNG]
set xpos [$rng uniform 2040 2160]
set ypos [$rng uniform 2040 2160]
$MH($i) set X_ $xpos
$MH($i) set Y_ $ypos
}
# MH starts to move towards to the FAs
$ns_ at 2.0 "$MH(0) setdest 2160.0 2160.0 11.1111"
$ns_ at [expr $dtime] "$MH(4) setdest 200.0 1.0 11.1111"
# create links between wired and BaseStation nodes
$ns_ duplex-link $W0  $HA  100Mb  25ms DropTail
$ns_ duplex-link $W0  $W00 100Mb   5ms DropTail
$ns_ duplex-link $W00 $CH1 100Mb  22ms DropTail
$ns_ duplex-link $W00 $CH2 100Mb  18ms DropTail
$ns_ duplex-link $W00 $CH3 100Mb  23ms DropTail
$ns_ duplex-link $W0  $W01 100Mb   5ms DropTail
$ns_ duplex-link $W01 $CH4 100Mb  22ms DropTail
$ns_ duplex-link $W01 $CH5 100Mb  18ms DropTail
$ns_ duplex-link $W01 $CH6 100Mb  23ms DropTail
$ns_ duplex-link $W0 $W1   100Mb 5ms DropTail
$ns_ duplex-link $W0 $W2   100Mb 5ms DropTail
$ns_ duplex-link $W0 $W3   100Mb 5ms DropTail
$ns_ duplex-link $W0 $W4   100Mb 5ms DropTail
$ns_ duplex-link $W1 $W5  100Mb 5ms DropTail
$ns_ duplex-link $W5 $W51 100Mb 5ms DropTail
$ns_ duplex-link $W5 $W52 100Mb 5ms DropTail
$ns_ duplex-link $W5 $W53 100Mb 5ms DropTail
$ns_ duplex-link $W5 $W54 100Mb 5ms DropTail
$ns_ duplex-link $W2 $W6  100Mb 5ms DropTail
$ns_ duplex-link $W6 $W61 100Mb 5ms DropTail
$ns_ duplex-link $W6 $W62 100Mb 5ms DropTail
$ns_ duplex-link $W6 $W63 100Mb 5ms DropTail
$ns_ duplex-link $W6 $W64 100Mb 5ms DropTail
$ns_ duplex-link $W3 $W7  100Mb 5ms DropTail
$ns_ duplex-link $W7 $W71 100Mb 5ms DropTail
$ns_ duplex-link $W7 $W72 100Mb 5ms DropTail
$ns_ duplex-link $W7 $W73 100Mb 5ms DropTail
$ns_ duplex-link $W7 $W74 100Mb 5ms DropTail
$ns_ duplex-link $W4 $W8  100Mb 5ms DropTail
$ns_ duplex-link $W8 $W81 100Mb 5ms DropTail
$ns_ duplex-link $W8 $W82 100Mb 5ms DropTail
$ns_ duplex-link $W8 $W83 100Mb 5ms DropTail
$ns_ duplex-link $W8 $W84 100Mb 5ms DropTail
# FAs
$ns_ duplex-link $W51 $FA(0) 100Mb 5ms DropTail
$ns_ duplex-link $W52 $FA(1) 100Mb 5ms DropTail
$ns_ duplex-link $W53 $FA(2) 100Mb 5ms DropTail
$ns_ duplex-link $W54 $FA(3) 100Mb 5ms DropTail
$ns_ duplex-link $W61 $FA(4) 100Mb 5ms DropTail
$ns_ duplex-link $W62 $FA(5) 100Mb 5ms DropTail
$ns_ duplex-link $W63 $FA(6) 100Mb 5ms DropTail
$ns_ duplex-link $W64 $FA(7) 100Mb 5ms DropTail
$ns_ duplex-link $W71 $FA(8) 100Mb 5ms DropTail
$ns_ duplex-link $W72 $FA(9) 100Mb 5ms DropTail
$ns_ duplex-link $W73 $FA(10) 100Mb 5ms DropTail
$ns_ duplex-link $W74 $FA(11) 100Mb 5ms DropTail
$ns_ duplex-link $W81 $FA(12) 100Mb 5ms DropTail
$ns_ duplex-link $W82 $FA(13) 100Mb 5ms DropTail
$ns_ duplex-link $W83 $FA(14) 100Mb 5ms DropTail
$ns_ duplex-link $W84 $FA(15) 100Mb 5ms DropTail

# Downlink CN --> MH
set tcp1 [new Agent/UDP]
$tcp1 set class_ 2
set sink1 [new Agent/LossMonitor]
```

```
    $ns_ attach-agent $MH(0) $tcp1
    $ns_ attach-agent $CH1 $sink1
    $ns_ connect $tcp1 $sink1
    set ftp1 [new Application/Traffic/CBR]
426 $ftp1 attach-agent $tcp1
    $ftp1 set type_ CBR
    $ftp1 set packetSize_ 500
    $ftp1 set interval_ 20ms
    $ns_ at 1.0 "$ftp1 start"
431 ## log lost packets
    $ns_ at 1.1 "record"
    $ns_ at 1.1 "record1"

    foreach i {5 10 15 20 25 30 35 40 50 55
436 60 65 70 75 80 85 90 95 100 105 110 115
    120 125 130 135 140 145 150 155} {
    set udp($i) [new Agent/UDP]
    #changed
    $udp($i) set packetSize_ 500
441 #/changed
    $ns_ attach-agent $CH1 $udp($i)
    set cbr_($i) [new Application/Traffic/CBR]
    $cbr_($i) set interval_ [expr $dtime]ms
    $cbr_($i) set packetSize_ 500
446 #$cbr_($i) set rate_ 0.2Mb
    #/changed
    $cbr_($i) attach-agent $udp($i)
    set null_($i) [new Agent/LossMonitor]
    $ns_ attach-agent $MH($i) $null_($i)
451 $ns_ connect $udp($i) $null_($i)
    $ns_ at 10.0 "$cbr_($i) start"
    $ns_ at 290.0 "$cbr_($i) stop"
    }
    #endof UDP
456 foreach i { 6 11 16 21 26 31 36 41 46 51
    56 61 66 71 76 81 86 91 96 101 106 111 116
    121 126 131 136 141 146 151} {
    set udp($i) [new Agent/UDP]
    #changed
461 $udp($i) set packetSize_ 500
    #/changed
    $ns_ attach-agent $CH5 $udp($i)
    set cbr_($i) [new Application/Traffic/CBR]
    $cbr_($i) set interval_ [expr $dtime]ms
466 $cbr_($i) set packetSize_ 500
    #$cbr_($i) set rate_ 0.2Mb
    #/changed
    $cbr_($i) attach-agent $udp($i)
    set null_($i) [new Agent/LossMonitor]
471 $ns_ attach-agent $MH($i) $null_($i)
    $ns_ connect $udp($i) $null_($i)
    $ns_ at 10.0  "$cbr_($i) start"
    $ns_ at 290.0 "$cbr_($i) stop"
    }
476 #endof UDP
    # Define initial node position in nam

    for {set i 0} {$i < $opt(nn)} {incr i} {

481     $ns_ initial_node_pos $node_($i) 20
    }
    # Tell all nodes when the siulation ends
    for {set i 0} {$i < $opt(nn) } {incr i} {
        $ns_ at $opt(stop).0 "$node_($i) reset";
486 }
    $ns_ at $opt(stop).0 "$HA reset";
    for {set l 0} {$l < 16} {incr l} {
    $ns_ at $opt(stop).0 "$FA($l) reset";
    }
491 $ns_ at $opt(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
    $ns_ at $opt(stop).0001 "stop"
    proc stop {} {
        global ns_ tracefd namtrace
        close $tracefd
```

```
496     close $namtrace
    }
    # write count of lost packets into lostfile
    proc record {} {
          global sink1 f0
501         #Get an instance of the simulator
            set ns_ [Simulator instance]
            set time 0.01
            set bw0 [$sink1 set nlost_]
            #Get the current time
506   set now [$ns_ now]
            puts $f0 "$now $bw0"
            #Reset the bytes_ values on the traffic sinks
            #Re-schedule the procedure
    $ns_ at [expr $now+$time] "record"
511 }
    # write count of packets into sendfile
    proc record1 {} {
      global sink1 f2
            #Get an instance of the simulator
516         set ns_ [Simulator instance]
      set time 0.01
      set pw0 [$sink1 set npkts_]
            #Get the current time
      set now [$ns_ now]
521         puts $f2 "$now $pw0"
            $ns_ at [expr $now+$time] "record1"
    }
    Agent/MIPMH instproc write_Operation {Operation_} {
            $self instvar node_
526         global f1
            set ns [Simulator instance]
            puts $f1 "$Operation_"
    }
    # some useful headers for tracefile
531 puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp \
            $opt(adhocRouting)"
    puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
    puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"
    puts "Starting Simulation..."
536 $ns_ run
```

A.2 HAWAII

Das folgende Tcl-Skript wurde für die Simulationen im Kapitel 6 des Protokolls HAWAII verwendet. Die Dateien *ns-wireless-mip.tcl* (ns-2 Bestandteil), *hawaii-lib.tcl* (Bestandteil des HAWAII-Patches [12]) müssen zusätzlich eingebunden werden.

```
    # HAWAII simulation in ns-2
    # Copyright (c) 2007 Technische Universität Ilmenau.
  3 # All rights reserved.
    # first author              : Christian Kellner
    # co-author and supervisor : Dipl.-Ing. Ali Diab

    set dtime [lindex $argv 0]
  8 puts "delay is $dtime ms"
    set sendfile [lindex $argv 1]
    puts "sendfile is $sendfile"
    set lostfile [lindex $argv 2]
    puts "lostfile is $lostfile"
 13
    #options for wireless domain
    set opt(chan)    Channel/WirelessChannel
    set opt(prop)    Propagation/TwoRayGround
    set opt(netif)   Phy/WirelessPhy
 18 set opt(mac)     Mac/802_11
    set opt(ifq)     Queue/DropTail/PriQueue
```

```
set opt(ll)             LL
set opt(ant)       Antenna/OmniAntenna
set opt(x)              2100
set opt(y)              2100
set opt(rp)             NOAH
set opt(ifqlen)         50
set opt(seed)           0.0
set opt(stop)           300.0
set opt(cc)             "off"
set opt(tr)             hawaii-air.tr
set opt(cp)             ""
set opt(sc)             ""
set opt(ftp0-start)     6.0
set opt(BgTrafficStart) 0.1
set num_wired_nodes     16
set num_bs_nodes        16
set num_wireless_nodes 161
set opt(nn) [expr $num_wireless_nodes + 20]

Agent/TCP set sport_            0
Agent/TCP set dport_            0
Agent/TCP set packetSize_       1460
Queue/DropTail/PriQueue set Prefer_Routing_Protocols 1
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 0.2
Antenna/OmniAntenna set Gr_ 0.2
Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6
Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0

source ns-wireless-mip.tcl

# Registration interval in Mobile
Agent/MIPMH set reg_rtx_ 1.0
Queue set limit_ 200
# number of Packets received in MH's sink
#set pktsNum 0
# intial setup - set addressing to hierarchical
set ns [new Simulator]
#$ns set-address-format hierarchical
$ns node-config -addressType hierarchical
# set mobileIP flag
Simulator set mobile_ip_ 1
# enable Hawaii Routing
set HawaiiRouting 1
# enable Hawaii MSF Routing
set HawaiiRoutingMSF 0
set namtrace [open /dev/null w]
$ns namtrace-all-wireless $namtrace $opt(x) $opt(y)
# suppress huge trace file
set trace [open /dev/null w]
$ns trace-all $trace
AddrParams set domain_num_ 1
lappend cluster_num 23
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 6 6 6 6 6 11 10 170 1 \
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
AddrParams set nodes_num_ $eilastlevel

set f2 [open $sendfile w]
set f0 [open $lostfile w]
set W0  [$ns node 0.0.0]
set W1  [$ns node 0.1.0]
set W5  [$ns node 0.1.1]
set W51 [$ns node 0.1.2]
set W52 [$ns node 0.1.3]
set W53 [$ns node 0.1.4]
set W54 [$ns node 0.1.5]
```

```
    set W2  [$ns node 0.2.0]
    set W6  [$ns node 0.2.1]
    set W61 [$ns node 0.2.2]
    set W62 [$ns node 0.2.3]
 98 set W63 [$ns node 0.2.4]
    set W64 [$ns node 0.2.5]
    set W3  [$ns node 0.3.0]
    set W7  [$ns node 0.3.1]
    set W71 [$ns node 0.3.2]
103 set W72 [$ns node 0.3.3]
    set W73 [$ns node 0.3.4]
    set W74 [$ns node 0.3.5]
    set W4  [$ns node 0.4.0]
    set W8  [$ns node 0.4.1]
108 set W81 [$ns node 0.4.2]
    set W82 [$ns node 0.4.3]
    set W83 [$ns node 0.4.4]
    set W84 [$ns node 0.4.5]
    set W00 [$ns node 0.5.0]
113 set W01 [$ns node 0.5.1]
    set CH1 [$ns node 0.5.2]
    set CH2 [$ns node 0.5.3]
    set CH3 [$ns node 0.5.4]
    set CH4 [$ns node 0.5.5]
118 set CH5 [$ns node 0.5.6]
    set CH6 [$ns node 0.5.7]
    set HA  [$ns node 0.5.8]
    $W0  color "maroon"
    $CH1 color "purple"
123 $CH2 color "purple"
    $CH3 color "purple"
    $CH4 color "purple"
    $CH5 color "purple"
    $CH6 color "purple"
128 $HA  color "red"
    # node W0 is the Domain Root Router
    set DRRaddress [$W0 address?]

    source hawaii-lib.tcl
133
    makeHawaiiRouter $W0  $DRRaddress
    makeHawaiiRouter $W1  $DRRaddress
    makeHawaiiRouter $W2  $DRRaddress
    makeHawaiiRouter $W3  $DRRaddress
138 makeHawaiiRouter $W4  $DRRaddress
    makeHawaiiRouter $W5  $DRRaddress
    makeHawaiiRouter $W6  $DRRaddress
    makeHawaiiRouter $W7  $DRRaddress
    makeHawaiiRouter $W8  $DRRaddress
143 makeHawaiiRouter $W51 $DRRaddress
    makeHawaiiRouter $W52 $DRRaddress
    makeHawaiiRouter $W53 $DRRaddress
    makeHawaiiRouter $W54 $DRRaddress
    makeHawaiiRouter $W61 $DRRaddress
148 makeHawaiiRouter $W62 $DRRaddress
    makeHawaiiRouter $W63 $DRRaddress
    makeHawaiiRouter $W64 $DRRaddress
    makeHawaiiRouter $W71 $DRRaddress
    makeHawaiiRouter $W72 $DRRaddress
153 makeHawaiiRouter $W73 $DRRaddress
    makeHawaiiRouter $W74 $DRRaddress
    makeHawaiiRouter $W81 $DRRaddress
    makeHawaiiRouter $W82 $DRRaddress
    makeHawaiiRouter $W83 $DRRaddress
158 makeHawaiiRouter $W84 $DRRaddress
    if { $opt(x) == 0 || $opt(y) == 0 } {
            puts "No X-Y boundary values given\n"
    }
    set chan        [new $opt(chan)]
163 set prop        [new $opt(prop)]
    set topo        [new Topography]
    # trace for CMUtrace, for wireless traffic
    set tracefd     [open /dev/null w]
    # setup topography and propagation model
```

```
$topo load_flatgrid $opt(x) $opt(y)
$prop topography $topo
# Create God
create-god $opt(nn)
# Configure using NOAH routing in Wireless domain
$ns node-config  -mobileIP ON \
                 -adhocRouting $opt(rp) \
                 -llType $opt(ll) \
                 -macType $opt(mac) \
                 -ifqType $opt(ifq) \
                 -ifqLen $opt(ifqlen) \
                 -antType $opt(ant) \
                 -propType $opt(prop) \
                 -phyType $opt(netif) \
                 -channelType $opt(chan) \
                 -topoInstance $topo \
                 -wiredRouting ON \
                 -agentTrace OFF \
                 -routerTrace OFF \
                 -macTrace OFF
set overlap 30
set power 0.29705643626340894
## setup Hawaii Base Station nodes using NOAH
$ns node-config -rxPower $power -txPower $power
set BS1 [$ns node 0.7.0]
$ns node-config -wiredRouting OFF;
for {set i 0} {$i < $num_wireless_nodes } {incr i} {
$ns node-config -rxPower $power -txPower $power
set MH_($i) [$ns node  0.7.[expr $i + 1]]
}
$ns node-config -wiredRouting ON;
## setup Hawaii Base Station nodes using NOAH
$ns node-config -rxPower $power -txPower $power
set BS2  [$ns node 0.8.0]
$ns node-config -rxPower $power -txPower $power
set BS3  [$ns node 0.9.0]
$ns node-config -rxPower $power -txPower $power
set BS4  [$ns node 0.10.0]
$ns node-config -rxPower $power -txPower $power
set BS5  [$ns node 0.11.0]
$ns node-config -rxPower $power -txPower $power
set BS6  [$ns node 0.12.0]
$ns node-config -rxPower $power -txPower $power
set BS7  [$ns node 0.13.0]
$ns node-config -rxPower $power -txPower $power
set BS8  [$ns node 0.14.0]
$ns node-config -rxPower $power -txPower $power
set BS9  [$ns node 0.15.0]
$ns node-config -rxPower $power -txPower $power
set BS10 [$ns node 0.16.0]
$ns node-config -rxPower $power -txPower $power
set BS11 [$ns node 0.17.0]
$ns node-config -rxPower $power -txPower $power
set BS12 [$ns node 0.18.0]
$ns node-config -rxPower $power -txPower $power
set BS13 [$ns node 0.19.0]
$ns node-config -rxPower $power -txPower $power
set BS14 [$ns node 0.20.0]
$ns node-config -rxPower $power -txPower $power
set BS15 [$ns node 0.21.0]
$ns node-config -rxPower $power -txPower $power
set BS16 [$ns node 0.22.0]
makeHawaiiBS    $BS1 $DRRaddress
createMsfBuffer $BS1 5 10
makeHawaiiBS    $BS2 $DRRaddress
createMsfBuffer $BS2 5 10
makeHawaiiBS    $BS3 $DRRaddress
createMsfBuffer $BS3 5 10
makeHawaiiBS    $BS4 $DRRaddress
createMsfBuffer $BS4 5 10
makeHawaiiBS    $BS5 $DRRaddress
createMsfBuffer $BS5 5 10
makeHawaiiBS    $BS6 $DRRaddress
createMsfBuffer $BS6 5 10
```

```
    makeHawaiiBS     $BS7 $DRRaddress
243 createMsfBuffer $BS7 5 10
    makeHawaiiBS     $BS8 $DRRaddress
    createMsfBuffer $BS8 5 10
    makeHawaiiBS     $BS9 $DRRaddress
    createMsfBuffer $BS9 5 10
248 makeHawaiiBS     $BS10 $DRRaddress
    createMsfBuffer $BS10 5 10
    makeHawaiiBS     $BS11 $DRRaddress
    createMsfBuffer $BS11 5 10
    makeHawaiiBS     $BS12 $DRRaddress
253 createMsfBuffer $BS12 5 10
    makeHawaiiBS     $BS13 $DRRaddress
    createMsfBuffer $BS13 5 10
    makeHawaiiBS     $BS14 $DRRaddress
    createMsfBuffer $BS14 5 10
258 makeHawaiiBS     $BS15 $DRRaddress
    createMsfBuffer $BS15 5 10
    makeHawaiiBS     $BS16 $DRRaddress
    createMsfBuffer $BS16 5 10
    $BS1 color "red"
263 $BS2 color "red"
    $BS3 color "red"
    $BS4 color "red"
    $BS5 color "red"
    $BS6 color "red"
268 $BS7 color "red"
    $BS8 color "red"
    $BS9 color "red"
    $BS10 color "red"
    $BS11 color "red"
273 $BS12 color "red"
    $BS13 color "red"
    $BS14 color "red"
    $BS15 color "red"
    $BS16 color "red"
278 ## observed MN is green; others are blue
    $MH_(0) color "green"
    $ns at 0.0 "$MH_(0) label MH_(0)"
    for {set i 1} {$i < $num_wireless_nodes } {incr i} {
    $MH_($i) color "blue"
283 $ns at 0.0 "$MH_($i) label MH_($i)"
    }
    $BS1 set X_ 1.000000000000
    $BS1 set Y_ 1.000000000000
    $BS1 set Z_ 0.000000000000
288 $BS2 set X_ 140.000000000000
    $BS2 set Y_ 140.000000000000
    $BS2 set Z_ 0.000000000000
    $BS3 set X_ 280.000000000000
    $BS3 set Y_ 280.000000000000
293 $BS3 set Z_ 0.000000000000
    $BS4 set X_ 420.000000000000
    $BS4 set Y_ 420.000000000000
    $BS4 set Z_ 0.000000000000
    $BS5 set X_ 560.000000000000
298 $BS5 set Y_ 560.000000000000
    $BS5 set Z_ 0.000000000000
    $BS6 set X_ 700.000000000000
    $BS6 set Y_ 700.000000000000
    $BS6 set Z_ 0.000000000000
303 $BS7 set X_ 840.000000000000
    $BS7 set Y_ 840.000000000000
    $BS7 set Z_ 0.000000000000
    $BS8 set X_ 980.000000000000
    $BS8 set Y_ 980.000000000000
308 $BS8 set Z_ 0.000000000000
    $BS9 set X_ 1120.000000000000
    $BS9 set Y_ 1120.000000000000
    $BS9 set Z_ 0.000000000000
    $BS10 set X_ 1260.000000000000
313 $BS10 set Y_ 1260.000000000000
    $BS10 set Z_ 0.000000000000
    $BS11 set X_ 1400.000000000000
```

```
    $BS11 set Y_ 1400.000000000000
    $BS11 set Z_ 0.000000000000
318 $BS12 set X_ 1540.000000000000
    $BS12 set Y_ 1540.000000000000
    $BS12 set Z_ 0.000000000000
    $BS13 set X_ 1680.000000000000
    $BS13 set Y_ 1680.000000000000
323 $BS13 set Z_ 0.000000000000
    $BS14 set X_ 1820.000000000000
    $BS14 set Y_ 1820.000000000000
    $BS14 set Z_ 0.000000000000
    $BS15 set X_ 1960.000000000000
328 $BS15 set Y_ 1960.000000000000
    $BS15 set Z_ 0.000000000000
    $BS16 set X_ 2100.000000000000
    $BS16 set Y_ 2100.000000000000
    $BS16 set Z_ 0.000000000000
333
    # Set default mobile movement speed
    set speed(1) 20.000000000000
    # Set tranmission Power
    set overlap 30
338 set BS1address [AddrParams addr2id [$BS1 node-addr]]
    set ns_ [Simulator instance]
    ## Label the Special Node in NAM
    $ns_ at 0.0 "$BS1 label BS1"
    $ns_ at 0.0 "$BS2 label BS2"
343 $ns_ at 0.0 "$BS3 label BS3"
    $ns_ at 0.0 "$BS4 label BS4"
    $ns_ at 0.0 "$BS5 label BS5"
    $ns_ at 0.0 "$BS6 label BS6"
    $ns_ at 0.0 "$BS7 label BS7"
348 $ns_ at 0.0 "$BS8 label BS8"
    $ns_ at 0.0 "$BS9 label BS9"
    $ns_ at 0.0 "$BS10 label BS10"
    $ns_ at 0.0 "$BS11 label BS11"
    $ns_ at 0.0 "$BS12 label BS12"
353 $ns_ at 0.0 "$BS13 label BS13"
    $ns_ at 0.0 "$BS14 label BS14"
    $ns_ at 0.0 "$BS15 label BS15"
    $ns_ at 0.0 "$BS16 label BS16"
    $ns_ at 0.0 "$W0  label DRR"
358 $ns_ at 0.0 "$W00  label W00"
    $ns_ at 0.0 "$W01  label W01"
    $ns_ at 0.0 "$W01  label W01"
    $ns_ at 0.0 "$CH1  label CH1"
    $ns_ at 0.0 "$CH2  label CH2"
363 $ns_ at 0.0 "$CH3  label CH3"
    $ns_ at 0.0 "$CH4  label CH4"
    $ns_ at 0.0 "$CH5  label CH5"
    $ns_ at 0.0 "$CH6  label CH6"
    $ns_ at 0.0 "$HA   label HA"
368 $ns_ at 0.0 "$W1  label W1"
    $ns_ at 0.0 "$W2  label W2"
    $ns_ at 0.0 "$W3  label W3"
    $ns_ at 0.0 "$W4  label W4"
    $ns_ at 0.0 "$W5  label W5"
373 $ns_ at 0.0 "$W51  label W51"
    $ns_ at 0.0 "$W52  label W52"
    $ns_ at 0.0 "$W53  label W53"
    $ns_ at 0.0 "$W54  label W54"
    $ns_ at 0.0 "$W6   label W6"
378 $ns_ at 0.0 "$W61  label W61"
    $ns_ at 0.0 "$W62  label W62"
    $ns_ at 0.0 "$W63  label W63"
    $ns_ at 0.0 "$W64  label W64"
    $ns_ at 0.0 "$W7   label W7"
383 $ns_ at 0.0 "$W71  label W71"
    $ns_ at 0.0 "$W72  label W72"
    $ns_ at 0.0 "$W73  label W73"
    $ns_ at 0.0 "$W74  label W74"
    $ns_ at 0.0 "$W8   label W8"
388 $ns_ at 0.0 "$W81  label W81"
    $ns_ at 0.0 "$W82  label W82"
```

```
    $ns_ at 0.0 "$W83  label W83"
    $ns_ at 0.0 "$W84  label W84"
    set NumOfHO 110
393 set stime 4
    set StayTime 2
    set spoint 10.0
    set dpoint 2060.0
    set dist [expr (sqrt(2 * (pow(($dpoint - $spoint), 2))))]
398 puts "**** DISTANCE: $dist ****"
    set trip_time [expr ($dist / $speed(1)) + $StayTime]
    $MH_(0) set Y_ $spoint
    $MH_(0) set X_ $spoint
    ## ten nodes at BS1
403 for {set i 1} {$i < 11} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 10 60]
    set ypos [$rng uniform 10 60]
    $MH_($i) set X_ $xpos
408 $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS2
    for {set i 11} {$i < 21} {incr i} {
    set rng [new RNG]
413 set xpos [$rng uniform 80 200]
    set ypos [$rng uniform 80 200]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
    }
418 ## ten nodes at BS3
    for {set i 21} {$i < 31} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 220 340]
    set ypos [$rng uniform 220 340]
423 $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS4
    for {set i 31} {$i < 41} {incr i} {
428 set rng [new RNG]
    set xpos [$rng uniform 360 480]
    set ypos [$rng uniform 360 480]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
433 }
    ## ten nodes at BS5
    for {set i 41} {$i < 51} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 500 620]
438 set ypos [$rng uniform 500 620]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS6
443 for {set i 51} {$i < 61} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 640 760]
    set ypos [$rng uniform 640 760]
    $MH_($i) set X_ $xpos
448 $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS7
    for {set i 61} {$i < 71} {incr i} {
    set rng [new RNG]
453 set xpos [$rng uniform 780 900]
    set ypos [$rng uniform 780 900]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
    }
458 ## ten nodes at BS8
    for {set i 71} {$i < 81} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 920 1040]
    set ypos [$rng uniform 920 1040]
463 $MH_($i) set X_ $xpos
```

```
    $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS9
    for {set i 81} {$i < 91} {incr i} {
468 set rng [new RNG]
    set xpos [$rng uniform 1060 1180]
    set ypos [$rng uniform 1060 1180]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
473 }
    ## ten nodes at BS10
    for {set i 91} {$i < 101} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 1200 1320]
478 set ypos [$rng uniform 1200 1320]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS11
483 for {set i 101} {$i < 111} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 1340 1460]
    set ypos [$rng uniform 1340 1460]
    $MH_($i) set X_ $xpos
488 $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS12
    for {set i 111} {$i < 121} {incr i} {
    set rng [new RNG]
493 set xpos [$rng uniform 1480 1600]
    set ypos [$rng uniform 1480 1600]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
    }
498 ## ten nodes at BS13
    for {set i 121} {$i < 131} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 1620 1740]
    set ypos [$rng uniform 1620 1740]
503 $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS14
    for {set i 131} {$i < 141} {incr i} {
508 set rng [new RNG]
    set xpos [$rng uniform 1760 1880]
    set ypos [$rng uniform 1760 1880]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
513 }
    ## ten nodes at BS15
    for {set i 141} {$i < 151} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 1900 2020]
518 set ypos [$rng uniform 1900 2020]
    $MH_($i) set X_ $xpos
    $MH_($i) set Y_ $ypos
    }
    ## ten nodes at BS16
523 for {set i 151} {$i < 161} {incr i} {
    set rng [new RNG]
    set xpos [$rng uniform 2040 2160]
    set ypos [$rng uniform 2040 2160]
    $MH_($i) set X_ $xpos
528 $MH_($i) set Y_ $ypos
    }
    $ns at 2.0 "$MH_(0) setdest $dpoint $dpoint 11.1111"
    if { $opt(cp) == "" } {
            #puts "*** NOTE: no connection pattern specified."
533         set opt(cp) "none"
    } else {
            puts "Loading connection pattern..."
            source $opt(cp)
    }
```

```
538 if { $opt(sc) == "" } {
        #puts "*** NOTE: no scenario file specified."
        set opt(sc) "none"
    } else {
        puts "Loading scenario file..."
543     source $opt(sc)
        puts "Load complete..."
    }
    # create links between wired and BaseStation nodes
    $ns duplex-link $CH1   $W00  100Mb 20ms DropTail
548 $ns duplex-link $CH2   $W00  100Mb 15ms DropTail
    $ns duplex-link $CH3   $W00  100Mb 12ms DropTail
    $ns duplex-link $CH4   $W01  100Mb 18ms DropTail
    $ns duplex-link $CH5   $W01  100Mb 22ms DropTail
    $ns duplex-link $CH6   $W01  100Mb 25ms DropTail
553 $ns duplex-link $W00   $W0  100Mb 5ms DropTail
    $ns duplex-link $W01   $W0  100Mb 5ms DropTail
    $ns duplex-link $W0   $W1  100Mb 5ms DropTail
    $ns duplex-link $W0   $W2  100Mb 5ms DropTail
    $ns duplex-link $W0   $W3  100Mb 5ms DropTail
558 $ns duplex-link $W0   $W4  100Mb 5ms DropTail
    $ns duplex-link $W0   $HA  100Mb 5ms DropTail
    $ns duplex-link $W1   $W5  100Mb 5ms DropTail
    $ns duplex-link $W5   $W51  100Mb 5ms DropTail
    $ns duplex-link $W5   $W52  100Mb 5ms DropTail
563 $ns duplex-link $W5   $W53  100Mb 5ms DropTail
    $ns duplex-link $W5   $W54  100Mb 5ms DropTail
    $ns duplex-link $W2   $W6  100Mb 5ms DropTail
    $ns duplex-link $W6   $W61  100Mb 5ms DropTail
    $ns duplex-link $W6   $W62  100Mb 5ms DropTail
568 $ns duplex-link $W6   $W63  100Mb 5ms DropTail
    $ns duplex-link $W6   $W64  100Mb 5ms DropTail
    $ns duplex-link $W3   $W7  100Mb 5ms DropTail
    $ns duplex-link $W7   $W71  100Mb 5ms DropTail
    $ns duplex-link $W7   $W72  100Mb 5ms DropTail
573 $ns duplex-link $W7   $W73  100Mb 5ms DropTail
    $ns duplex-link $W7   $W74  100Mb 5ms DropTail
    $ns duplex-link $W4   $W8  100Mb 5ms DropTail
    $ns duplex-link $W8   $W81  100Mb 5ms DropTail
    $ns duplex-link $W8   $W82  100Mb 5ms DropTail
578 $ns duplex-link $W8   $W83  100Mb 5ms DropTail
    $ns duplex-link $W8   $W84  100Mb 5ms DropTail
    $ns duplex-link $W51 $BS1  100Mb 5ms DropTail
    $ns duplex-link $W52 $BS2  100Mb 5ms DropTail
    $ns duplex-link $W53 $BS3  100Mb 5ms DropTail
583 $ns duplex-link $W54 $BS4  100Mb 5ms DropTail
    $ns duplex-link $W61 $BS5  100Mb 5ms DropTail
    $ns duplex-link $W62 $BS6  100Mb 5ms DropTail
    $ns duplex-link $W63 $BS7  100Mb 5ms DropTail
    $ns duplex-link $W64 $BS8  100Mb 5ms DropTail
588 $ns duplex-link $W71 $BS9  100Mb 5ms DropTail
    $ns duplex-link $W72 $BS10  100Mb 5ms DropTail
    $ns duplex-link $W73 $BS11  100Mb 5ms DropTail
    $ns duplex-link $W74 $BS12  100Mb 5ms DropTail
    $ns duplex-link $W81 $BS13  100Mb 5ms DropTail
593 $ns duplex-link $W82 $BS14  100Mb 5ms DropTail
    $ns duplex-link $W83 $BS15  100Mb 5ms DropTail
    $ns duplex-link $W84 $BS16  100Mb 5ms DropTail
    $ns color 1 Blue
    $ns color 5 Blue
598 $ns color 6 Blue
    $ns color 7 Blue
    $ns color 8 green
    $ns color 22 Blue
    $ns color 0 blue
603 for {set i 0} {$i < 1} {incr i} {
    set udp($i) [new Agent/UDP]
    $udp($i) set packetSize_ 500
    $ns attach-agent $CH1 $udp($i)
    set cbr_($i) [new Application/Traffic/CBR]
608 $cbr_($i) set interval_ 20ms
    $cbr_($i) set packetSize_ 500
    $cbr_($i) attach-agent $udp($i)
    set null_($i) [new Agent/LossMonitor]
```

```
    $ns attach-agent $MH_($i) $null_($i)
613 $ns connect $udp($i) $null_($i)
    $ns at 1.0 "$cbr_($i) start"
    ## log lost packets
    $ns_ at 1.1 "record"
    $ns_ at 1.1 "record1"
618 $ns at [expr $opt(stop) - 0.5] "$cbr_($i) stop"
    }
    #endof UDP
    foreach i {5 10 15 20 25 30 35 40 50 55
    60 65 70 75 80 85 90 95 100 105 110 115
623 120 125 130 135 140 145 150 155} {
    set udp($i) [new Agent/UDP]
    $udp($i) set packetSize_ 500
    $ns attach-agent $CH2 $udp($i)
    set cbr_($i) [new Application/Traffic/CBR]
628 $cbr_($i) set interval_ [expr $dtime]ms
    $cbr_($i) set packetSize_ 500
    $cbr_($i) attach-agent $udp($i)
    set null_($i) [new Agent/LossMonitor]
    $ns attach-agent $MH_($i) $null_($i)
633 $ns connect $udp($i) $null_($i)
    $ns at 10.0 "$cbr_($i) start"
    $ns at 290.0 "$cbr_($i) stop"
    }
    #endof UDP
638 foreach i { 6 11 16 21 26 31 36 41 46 51
    56 61 66 71 76 81 86 91 96 101 106 111 116
    121 126 131 136 141 146 151} {
    set udp($i) [new Agent/UDP]
    $udp($i) set packetSize_ 500
643 $ns attach-agent $CH5 $udp($i)
    set cbr_($i) [new Application/Traffic/CBR]
    $cbr_($i) set interval_ [expr $dtime]ms
    $cbr_($i) set packetSize_ 500
    $cbr_($i) attach-agent $udp($i)
648 set null_($i) [new Agent/LossMonitor]
    $ns attach-agent $MH_($i) $null_($i)
    $ns connect $udp($i) $null_($i)
    $ns at 10.0  "$cbr_($i) start"
    $ns at 290.0 "$cbr_($i) stop"
653 }
    #endof UDP
    for {set i 0} {$i < $num_wireless_nodes } {incr i} {
        $ns_ at $opt(stop).0000010 "$MH_($i) reset";
    }
658
    $ns_ at $opt(stop).0000010 "$BS1 reset";
    $ns_ at $opt(stop).0000010 "$BS2 reset";
    $ns_ at $opt(stop).0000010 "$BS3 reset";
    $ns_ at $opt(stop).0000010 "$BS4 reset";
663 $ns_ at $opt(stop).0000010 "$BS5 reset";
    $ns_ at $opt(stop).0000010 "$BS6 reset";
    $ns_ at $opt(stop).0000010 "$BS7 reset";
    $ns_ at $opt(stop).0000010 "$BS8 reset";
    $ns_ at $opt(stop).0000010 "$BS9 reset";
668 $ns_ at $opt(stop).0000010 "$BS10 reset";
    $ns_ at $opt(stop).0000010 "$BS11 reset";
    $ns_ at $opt(stop).0000010 "$BS12 reset";
    $ns_ at $opt(stop).0000010 "$BS13 reset";
    $ns_ at $opt(stop).0000010 "$BS14 reset";
673 $ns_ at $opt(stop).0000010 "$BS15 reset";
    $ns_ at $opt(stop).0000010 "$BS16 reset";
    $ns_ at $opt(stop).21 "finish"
    $ns_ at $opt(stop).20 "puts \"NS EXITING...\" ; "

678 proc finish {} {
        global ns_ trace namtrace null_ cbr_ pktsNum mytrace
        global HawaiiRoutingMSF
        if { $HawaiiRoutingMSF == 1 } {
            puts "Result for HAWAII using MSF scheme"
683     } else {
            puts "Result for HAWAII using UNF scheme"
        }
```

```
    foreach i {0} {
          puts "Total number of packet lost:\
688 [expr [$cbr_($i) set seqno_]-\
    [$null_($i) set npkts_]]"
    puts "Total packet sent:[$cbr_($i) set seqno_] \
          received:[$null_($i) set npkts_]"
     }
693       $ns_ flush-trace
          close $namtrace
          close $trace
          puts "Finishing ns.."
          exit 0
698 }
    # write count of lost packets into lostfile
    proc record {} {
          global null_ f0
            #Get an instance of the simulator
703         set ns_ [Simulator instance]
            set time 0.01
            set bw0 [$null_(0) set nlost_]
            #Get the current time
          set now [$ns_ now]
708         puts $f0 "$now $bw0"
            #Reset the bytes_ values on the traffic sinks
          $ns_ at [expr $now+$time] "record"
    }
    # write count of packets into sendfile
713 proc record1 {} {
          global null_ f2
            #Get an instance of the simulator
            set ns_ [Simulator instance]
            set time 0.01
718         set pw0 [$null_(0) set npkts_]
            #Get the current time
          set now [$ns_ now]
            puts $f2 "$now $pw0"
          $ns_ at [expr $now+$time] "record1"
723 }
    puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp $opt(rp)"
    puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
    puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"
    puts "Starting Simulation..."
728 $ns_ run
```

B Quellcode L-MIFA

In diesem Abschnitt sind die wichtigsten Funktionen für die Implementierung von L-MIFA hinterlegt. Alle veränderten Routinen werden in die Datei *mip-reg.cc* eingearbeitet.

```
void MIPBSAgent::sendOutMessagetoMN(int coa_,
int ha_,int haddr_, int pfa_, int dest_addr_, MifaRegType type_,
int seq,double lifetime_, const char *Message_,const char *ID_)
 {
        Tcl& tcl = Tcl::instance(); // TCL-Instanz
        Packet *pkt = allocpkt(); // Erzeugung eines Paketes
        hdr_ip *iph_ = hdr_ip::access(pkt); // Zugriff auf IP-Paket Header
        hdr_mip *h_ = hdr_mip::access(pkt); // Zugriff auf MIP-Header
        hdr_cmn *ch_ = hdr_cmn::access(pkt);
        h_->type_ = type_;
        h_->lifetime_=lifetime_;
        h_->coa_=coa_;
        h_->ha_=ha_;
        h_->haddr_=haddr_;
        h_->ID_=ID_;
        h_->pfa_=pfa_;
        h_->MIFA_Flag=1;
        iph_->saddr() = Address::instance().get_nodeaddr(addr());
        iph_->sport() = port();
        iph_->daddr() = dest_addr_; // Zieladresse: prev. FA
        iph_->dport() = 0; // Zielport
        NsObject *target = (NsObject *)tcl.lookup(tcl.result());
        if (target != NULL) {
          ((NsObject *)tcl.lookup(tcl.result()))->recv(pkt, (Handler*) 0);
          }
        else {
        target=(NsObject *)tcl.lookup(ID_);
        if (target != NULL) {
          target->recv(pkt, (Handler*) 0);
          }
          else
          send(pkt, 0);
          }
}
void MIPBSAgent::sendOutControlMessage(int coa_,int ha_,
int haddr_, int pfa_,int dest_addr_, MifaRegType type_,
int seq,double lifetime_, const char *Message_,const char *ID_)
{
        Tcl& tcl = Tcl::instance(); // TCL-Instanz
        Packet *pkt = allocpkt(); // Erzeugung eines Paketes
        hdr_ip *iph_ = hdr_ip::access(pkt); // Zugriff auf IP-Paket Header
        hdr_mip *h_ = hdr_mip::access(pkt); // Zugriff auf MIP-Header
        hdr_cmn *ch_ = hdr_cmn::access(pkt);
        h_->type_ = type_;
        h_->lifetime_=lifetime_;
        h_->coa_=coa_;
        h_->ha_=ha_;
        h_->haddr_=haddr_;
        h_->ID_=ID_;
        h_->pfa_=pfa_;
        h_->MIFA_Flag=1;
        iph_->saddr() = Address::instance().get_nodeaddr(addr());
        iph_->sport() = port();
        iph_->daddr() = dest_addr_; // Zieladresse: prev. FA
```

```
        iph_->dport() = 0; // Zielport
        send(pkt, 0);
}
void MIPBSAgent::recv(Packet* p, Handler *)
{
        Tcl& tcl = Tcl::instance();
        char out_Op[200];
        char str[100];
        const char *objname = NULL;
        NsObject *obj = NULL;
        hdr_mip *miph = hdr_mip::access(p);
        hdr_ip *iph = hdr_ip::access(p);
        hdr_cmn *ch = hdr_cmn::access(p);
        int nodeaddr = Address::instance().get_nodeaddr(addr());
        int dest_;// variable passed to Encapsulator
        switch (miph->type_) {

        case PRSOL:
                sendOutMessagetoMN(miph->coa_, miph->ha_, miph->haddr_,
                miph->pfa_, miph->haddr_, PRADV, miph->seqno_,miph->lifetime_,
                "PRADV",miph->ID_);
                break;

        case REGREQ1:
                sendOutMessagetoMN(miph->coa_, miph->ha_, miph->haddr_, miph->pfa_,
                iph->saddr(), IntAck, miph->seqno_,miph->lifetime_, "IntAck",miph->ID_);
                sendOutControlMessage(miph->coa_, miph->ha_, miph->haddr_, miph->pfa_,
                miph->coa_, REGREQ2, miph->seqno_,miph->lifetime_, "REGREQ2",miph->ID_);
                break;

        case REGREQ2:
                sendOutControlMessage(miph->coa_, miph->ha_, miph->haddr_, miph->pfa_,
                iph->saddr(), REGREPLY1, miph->seqno_, miph->lifetime_, "REGREPLY1",miph->ID_);
                sendOutControlMessage(miph->coa_, miph->ha_, miph->haddr_, miph->pfa_, miph->ha_,
                HA_NOT, miph->seqno_,miph->lifetime_, "HA_NOT",miph->ID_);
                latency_Time=Scheduler::instance().clock();
                sendOutMessagetoMN(miph->coa_, miph->ha_, miph->haddr_, miph->pfa_, miph->haddr_,
                REGREPLY2, miph->seqno_,                    miph->lifetime_, "REGREPLY2",miph->ID_);
                obj = (NsObject*)tcl.lookup(objname = tcl.result());
                if (strlen(objname) == 0)
                objname = "XXX";
                if (miph->ha_!=Address::instance().get_nodeaddr(addr())) {
                        tcl.evalf("s decap-route d s lf", name_, miph->haddr_,
                  objname, miph->lifetime_);
                }
                break;

        case HA_NOT:
                sendOutControlMessage(miph->coa_, miph->ha_, miph->coa_, miph->pfa_,
                iph->saddr(), HA_ACK, miph->seqno_, miph->lifetime_, "HA_ACK",miph->ID_);
                if (miph->coa_>0 && miph->coa_!=miph->ha_) {
                dest_=miph->coa_;
                EncapsFull_=1;
                tcl.evalf("s encap-route d d lf d d d d", name_,
                        miph->haddr_, miph->coa_, miph->lifetime_,EncapsFull_,
                        0, miph->haddr_,MN_Served);
                MN_Served++;
                }
                break;

        case HA_ACK:
                break;

        case REGREPLY1:
                if (miph->pfa_!=miph->coa_) {
                if (miph->coa_>0 && miph->coa_!=miph->ha_) {
                dest_=miph->coa_;
                objname = "XXX";
                if (miph->ha_!=Address::instance().get_nodeaddr(addr())) {
                // decapsulation
                tcl.evalf("s decap-route d d lf d d d d", name_,
                miph->haddr_, objname, miph->coa_,miph->lifetime_,EncapsFull_,
                0, miph->haddr_,MN_Served);
                MN_Served++;
```

```
                }}}
                break;

        case REGREPLY2:
                break;
        default:
                break;
        }
        Packet::free(p);

        }
}
void MIPMHAgent::sendOutControlMessage(int coa_,int dest_addr_,
MifaRegType type_, int seq,double lifetime_, const char *Message_,
const char *ID_)
{
        Tcl& tcl = Tcl::instance(); // TCL-Instanz
        Packet *pkt = allocpkt(); // Erzeugung eines Paketes
        hdr_ip *iph_ = hdr_ip::access(pkt); // Zugriff auf IP-Paket Header
        hdr_mip *h_ = hdr_mip::access(pkt); // Zugriff auf MIP-Header
        hdr_cmn *ch_ = hdr_cmn::access(pkt);
        h_->type_ = type_;
        h_->lifetime_=lifetime_;
        h_->coa_=coa_;
        h_->ha_=ha_;
        h_->ID_=ID_;
        h_->MIFA_Flag=1;
        h_->haddr_=Address::instance().get_nodeaddr(addr());
        iph_->saddr() = Address::instance().get_nodeaddr(addr());
        iph_->sport() = port();
        iph_->daddr() = dest_addr_; // Zieladresse: prev. FA
        iph_->dport() = 0; // Zielport
        send(pkt,0);
}
void MIPMHAgent::recv(Packet* p, Handler *)
{
        Tcl& tcl = Tcl::instance();
        hdr_mip *miph = hdr_mip::access(p);
        hdr_ip *iph = hdr_ip::access(p);
        char out_Op[300];
        char str[200];
        switch (miph->type_) {
        case MIPT_ADS:
        {
          Sending_Time=Scheduler::instance().clock();
            AgentList **ppagts = &agts_, *ptr;
            while (*ppagts) {
              if ((*ppagts)->node_ == miph->coa_) break;
              ppagts = &(*ppagts)->next_;
            }
            if (*ppagts) {
              ptr = *ppagts;
              *ppagts = ptr->next_;
              ptr->expire_time_ = beacon_ +
                Scheduler::instance().clock();
              ptr->lifetime_ = miph->lifetime_;
              ptr->next_ = agts_;
              agts_ = ptr;
              if (agts_->next_)
         if (miph->MIFA_Flag==0) {
              if (coa_ == miph->coa_) {
                seqno_++;
                }
              else {
              prob=false; // to register new advertisement
              counter=0;
              }
         }
         else {
        // L-MIFA
               if (coa_ == miph->coa_) {
                 seqno_++;
              }
              else {
```

```
                prob=false; //only register new advertisement
                counter=0;
                } } }
              else { // new ads
207             ptr = new AgentList;
                ptr->node_ = miph->coa_;
                ptr->expire_time_ = beacon_ + ClockTime;
                ptr->lifetime_ = miph->lifetime_;
                ptr->next_ = agts_;
212             agts_ = ptr;
                if (agts_->next_)
                if (miph->MIFA_Flag==1) {
                if (coa_>0) {
                if (pfa_<0)
217             pfa_=coa_;
                if (coa_!= miph->coa_)
                pfa_=coa_;
                 if (miph->coa_==ha_)
                pfa_=ha_;
222                 } else{
                  pfa_=miph->coa_;
                  }
                   }
                  if (pfa_<0)
227               pfa_=miph->coa_;
                coa_ = miph->coa_;
                Tcl& tcl = Tcl::instance();
                Packet *p = allocpkt();
                  sendOutControlMessage(coa_, pfa_, PRSOL, seqno_,
232               miph->lifetime_, "PRSOL",name_);
                if(node_)
                  node_->set_base_stn(coa_);
                  adlftm_ = miph->lifetime_;
                  seqno_++;
237               prob=false; //only register new advertisement
                  counter=0;
             }
            break;
            }
242       case PRADV:
                  reg();
                  break;
          case REGREPLY2:
          if(Pcoa_ != miph->coa_) {
247       if (miph->haddr_ == 4194305) {
          // here we put expression for evaluation
          }}
          Pcoa_ = miph->coa_;
                  break;
252       default:
            Packet::free(p);
            break;
    }}
```

Literaturverzeichnis

[1] Heise-Online: „Computex: VoIP-Telefon mit WLAN“, http://www.heise.de/newsticker/meldung/60223

[2] innovations report: „WLAN-Hotspot-Boom in Europa“, http://www.innovations-report.de/html/berichte/studien/bericht-23501.html

[3] Chung, Claypool: „NS by Example“, Worcester Polytechnic Institute, http://nile.wpi.edu/NS/

[4] The ns Manual. http://www.isi.edu/nsnam/ns/ns-documentation.html, 5. Februar 2007

[5] T. Campbell, J. Gomez, S. Kim, Chieh-Yih Wan, R. Turanyi, G. Valko: „Comparison Of IP Micromobility Protocols“, http://www.comet.columbia.edu/cellularip/pub/pcs2001.pdf IEEE Wireless Communications, Feburary 2002

[6] C. Perkins: „IP Mobility Support for IPv4“, RFC 3344, Network Working Group, http://tools.ietf.org/html/rfc3344, August 2002

[7] E. Gustafsson, A. Jonsson, C. Perkins: „Mobile IP Regional Registration“, Internet Draft, IETF 2001, http://comet.columbia.edu/micromobility/pub/draft-ietf-mobileip-reg-tunnel-04.txt

[8] P. Reinbold, O. Bonaventure: „IP Micro-mobility Protocols“ IEEE Communications Surveys and Tutorials 5 2003, http://www.comsoc.org/livepubs/surveys/public/2003/sep/reinbold.html

[9] G. Valko, T. Campbell, J. Gomez, Chieh-Yih Wan, S. Kim, R. Turanyi: „Celluar IP“, IETF Internet Draft Mai 1999, http://comet.columbia.edu/cellularip/pub/draft-valko-cellularip-00.txt

[10] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, L. Salgarelli: „IP micro-mobility support using HAWAII“, IETF Internet Draft Juni 1999, http://tools.ietf.org/html/draft-ramjee-micro-mobility-hawaii-00.txt

[11] D. Johnson, C. Perkins, J. Arkko: „Mobility Support in IPv6“, RFC3375, Network Working Group, Juni 2004, http://www.ietf.org/rfc/rfc3775.txt

[12] Columbia IP Micro-Mobility Suite (CIMS) updated to ns-2.29, URL: http://wcms1.rz.tu-ilmenau.de/fakia/Micro-Mobility.5233.0.html

[13] A. Diab, A. Mitschele-Thiel, E. Al Nasouri, R. Böringer, J. Xu: „Mobile IP Fast Authentication Protocol“, Technische Universität Ilmenau, Fachgebiet Integrierte HW/SW-Systeme

[14] A. Diab, A. Mitschele-Thiel: „Minimizing Mobile IP Handoff Latency“, Technische Universität Ilmenau, Fachgebiet Integrierte HW/SW-Systeme

[15] S.K. Sen, et al: „A Selective Location Update Strategy for PCS Users.“ ACM/-Baltzer J. Wireless Networks, September 1999

[16] K. El Malki: „Low Latency Handoffs in Mobile IPv4“, IETF Internet Draft April 2006, http://www.ietf.org/rfc/rfc4881.txt

[17] A. Diab, A. Mitschele-Thiel and R. Böringer: „Extension of Mobile IP for Fast Authentication and Low Latency Handoff“, Technische Universität Ilmenau, Fachgebiet Integrierte HW/SW-Systeme, Februar 2005

[18] M. Dunmore: „Mobile IPv6 Handovers: Performance Analysis and Evaluation“, Juni 2005

[19] I. Samprakou, C. J. Bouras, T. Karoubalis: „Improvements on „IP-IAPP“: A fast handoff protocol for IEEE 802.11 wireless and mobile clients“, Springer+Business Media, Juni 2006

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AAA ..... **A**uthentication, **A**uthorization and **A**ccount
AFA ..... **A**nchor **F**oreign **A**gent
AP ..... **A**ccess **P**oint
CCoA ..... **Co**-located **C**are-**o**f **A**ddress
CIMS ..... **C**olumbia **I**P **M**icro-Mobility **S**oftware
CIP ..... **C**elluar **I**P **P**rotocol
CN ..... **C**orrespondent **N**ode
CoA ..... **C**are-**o**f **A**ddress
DRR ..... **D**omain, **R**oot and **R**outer
ECS ..... **E**ager **C**ell **S**witching
FA ..... **F**oreign **A**gent
FHR ..... **F**requent **H**andoff **R**egion
GFA ..... **G**ateway **F**oreign **A**gent
HA ..... **H**ome **A**gent
HAWAII ..... **H**andoff-**A**ware **W**ireless **A**ccess **I**nternet **I**nfrastructure
HMIP ..... **H**ierarchical **M**obile **I**P **P**rotocol
ICMP ..... **I**nternet **C**ontrol **M**essage **P**rotocol
IEEE ..... **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IETF ..... **I**nternet **E**ngineering **T**ask **F**orce
IPTV ..... **I**nternet **P**rotocol **Tel**e**v**ision
ISP ..... **I**nternet **S**ervice **P**rovider
L-MIFA ..... **L**ow Latency **M**obile **I**P **F**ast **A**uthentication Protocol
L2-LD ..... **L**ayer **2**-**L**ink **D**own
L2-LU ..... **L**ayer **2**-**L**ink **U**p
L3-FHR ..... **L**ayer **3**-**F**requent **H**andoff **R**egion
LCS ..... **L**azy **C**ell **S**witching
MA ..... **M**obility **A**gent

MH ..... **M**obile **H**ost
MIFA ..... **M**obile **I**P **F**ast **A**uthentication Protocol
MN ..... **M**obile **N**ode
MRP ..... **M**obile **R**outing **P**oint
nFA ..... **n**ew **F**oreign **A**gent
oFA ..... **o**ld **F**oreign **A**gent
PFANE ..... **P**revious **F**oreign **A**gent **N**otification **E**xtension
QoS ..... **Q**uality **o**f **S**ervice
RFA ..... **R**egional **F**oreign **A**gent
RTT ..... **R**ound **T**rip **T**ime
SDL ..... **S**pecification and **D**escription **L**anguage

Thesen zur Diplomarbeit

1. Ziel der Arbeit war die Implementierung und Evaluierung von L-MIFA.
2. Die Performanz von MIP, HAWAII, MIFA und L-MIFA wurde analysiert. Ebenfalls untersucht wurde, wie stark sich die Änderung der Last auf die Performanz der Protokolle auswirkt.
3. Die Analysen haben gezeigt, dass die Handoff Latenzzeiten und Paketverlustraten von MIFA und L-MIFA deutlich geringer sind als bei HAWAII und MIP. Bei L-MIFA liegen diese Werte nahe Null.
4. Des weiteren haben die Analysen ergeben, das die Änderung der Last sich auf MIFA und L-MIFA nur geringfügig auswirken, während dies bei HAWAII und MIP einen starken Anstieg der Handoff Latenzzeiten und Paketverlustraten bewirkt.
5. MIFA und L-MIFA erfüllen mit einer durchschnittlichen Latenzzeit von deutlich unter 50 ms Echtzeitanforderungen.

Ilmenau, den 16. 07. 2007 Christian Kellner

Erklärung

Die vorliegende Arbeit habe ich selbstständig ohne Benutzung anderer als der angegebenen Quellen angefertigt. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Quellen entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer oder anderer Prüfungen noch nicht vorgelegt worden.

Ilmenau, den 16. 07. 2007 Christian Kellner