FACHHOCHSCHULE FURTWANGEN
HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT

FACHBEREICH INFORMATIONSSYSTEME
STUDIENGANG COMPUTER NETWORKING

PROJEKTSTUDIUM 3 · SS 2003

MICHAEL EYER · EYER@FOO.FH-FURTWANGEN.DE
ROGER KLOSE · ROGERKLOSE@WEB.DE
MARCIN LADECKI · MARCIN@LADECKI.NET
DANIEL MÜLLER · DANIELM79@GMX.DE
ADRIAN WOIZIK · ADRIAN@WOIZIK.DE

ERFORSCHUNG DES DERZEITIGEN ENTWICKLUNGSSTANDES BEI MOBILE IPV6 ENTWICKLUNGEN

Abstract

Die vorliegende Arbeit ist die schriftliche Dokumentation des im Sommersemester 2003 durchgeführten Semesterprojekts im Studiengangs „Computer Networking“ an der Fachhochschule Furtwangen.

Dieses Dokument wurde unter Verwendung von LaTeX und BiBTeX erstellt.

Furtwangen, im Juli 2003

Michael Eyer
Roger Klose
Marcin Ladecki
Daniel Müller
Adrian Woizik

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1
Einleitung

Im Studiengang „Computer Networking“ der Fachhochschule Furtwangen ist sowohl im 5. als auch im 7. Semester ein „Projektstudium“ vorgesehen. Innerhalb dieses Projektstudiums soll im Laufe des Semesters ein grösseres Projekt in Teamarbeit geplant, durchgeführt und dokumentiert werden.

Bei der Durchführung des Projektes spielen neben dem rein technischen Aspekt auch Themen wie Teamarbeit und Projektmanagement eine Rolle.

1.1 Projektziel

Ziel des Projektes ist die Evaluierung von Mobile IPv6 Implementationen und dessen derzeitigem Entwicklungsstandes. Sollte eine erfolgreiche Implementation vorhanden sein soll diese in ein stabiles Versuchsfeld integriert werden.

Für die Durchführung des Projektes sind folgende Aufgaben notwendig:

- Installation der Notebooks
- Patchen des Kernels auf Mobile IPv6 Funktionalität
- Einbinden der WLAN-Karten
- Aufsetzen & Patchen des HomeAgents
- Aufsetzen & Patchen des Corresponding Nodes
- Einbinden der Access Points
- Strukturierung des Netzwerkes
- Anpassung der Konfigurations Dateien
- Fehleranalyse
- Re-Konfiguration
- Performaneanalyse

- Einbindung des Fast-Handover
- Erhöhung der Verfügbarkeit des Mobile IPv6 Systems
- Abschlusstests

1.2 Ausgangssituation

Der Studiengang Computer-Networking wurde von der Firma Hewlett Packard GmBH gebeten ein Projekt durchzuführen welches sich mit dem derzeitigem Stand von Mobile-IPv6 beschäftigt. Hierzu stellte Hewlett-Packard unserer Projekt Gruppe zwei Compaq evo610c Notebooks mit integriertem USB-WLAN zur Verfügung. Mit den uns zu Verfügung stehenden Mitteln sollte dann ein Testszenario erstellt werden in welchem die Mobile-IPv6 Funktionalität analysiert werden konnte. Unsere Gruppe besaß bereits zwei Access-Points die man jeweils mit einem Gateway verbinden konnte. Diese Gateways wurden auch aus eigener Hardware zusammengesetzt, bzw. verwendet. Die Vorhandene Infrastruktur in der Fachochschule machte es uns unmöglich vor Ort zu arbeiten, deshalb wurde das Gesammte Projekt in das Studentenwohnheim Am Grosshausberg 9 umgezogen. FIXME: WAS HINZUGEFUEGT, REICHT DAS NU?

1.3 Dokumentstruktur

FIXME: ADD DOKUSTRUCT

1.4 Grundlagen

Es ist bereits rund dreißig Jahre her, dass das Internet Protokoll Version 4 (IPv4) eingeführt wurde. In dieser langen Zeit haben sich viele Anforderungen an das Internet geändert. IPv4 wird diesen Anforderungen nicht mehr gerecht und mobile, internetbasierte Multimedia-Dienste können erst dann vollkommen bereitgestellt werden, wenn das Internet verbessert wird. Das größte Problem dabei ist wohl der gewaltige Aufschwung, also die stetig steigende Benutzerzahl, des Internets. Nach einer Schätzung der Internet Engineering Task Force (IETF) wird es etwa im Jahr 2010 keine freien IP-Adressen mehr geben. Diese Schätzungen beziehen sich aber nur auf PC basierte Internetanschlüsse. Wenn man die Einführung von UMTS betrachtet und somit die steigende Anzahl mobiler Endgeräte mit IP-Anschluss einbezieht, wird der Vorrat an IP-Adressen wohl schon viel eher aufgebraucht sein. Nach Angaben und Prognosen der Europäischen Kommission wird der Anteil der Mobilfunkbenutzer an der Gesamtbevölkerung im Jahr 2003 auf 65% steigen.

Bereits seit Anfang der 90er Jahre arbeitet die IETF an einem Nachfolger für IPv4. Zwischen 1995 und 1996 wurde aus verschiedenen Entwürfen zum Internet Protokoll next Generation (IPnG) ein einheitlicher Standard entwickelt. Dieses Protokoll ist aber eher unter dem Namen IPv6 bekannt.

Das Wort Mobilität gewinnt im Informationszeitalter immer mehr an Bedeutung. Mobilität wurde bereits durch ein zusätzliches Protokoll realisiert, dem Mobile IPv4. Dieses Protokoll konnte aber aufgrund einiger Schwächen nicht wirklich überzeugen. Bei der Entwicklung von IPv6 legte man also auch großes Augenmerk auf die Unterstützung von mobilen Geräten und entwickelte den M̈obility Support in “IPv6”.

Kapitel 2
Mobile IPv6 Technologien

2.1 Einführung

Dieses Kapitel befasst sich mit den Grundlagen eines Mobile IPv6 Netzes. Wir werden hauptsächlich auf die Makrobeweglichkeitsmechanismen eingehen. Die mobilen Knotenbewegungen innerhalb von Funknetzen wie z. B. bei der Mobilfunktelephonie werden wir hier nicht behandeln. Die hier beschriebenen Mobile-IP-Lösungen befassen sich mit Makrobeweglichkeitsmechanismen auf Layer 3. Der Lösungszweck zielt folgende Punkte an:

- Die Aufrechterhaltung der Kommunikation zwischen einem mobilen Knoten und einem korrespondentierenden Knoten während der mobile Knoten sich von einem Subnetz in ein anderes Subnetz bewegt. Solch ein Mechanismus sollte so nahtlos wie möglich sein.
- Eine Verbindung mit einem mobilen Knoten sollte immer mit der gleichen IP-Adresse möglich sein, egal in welchem Subnetz er sich gerade befindet.

2.2 Mobiles IPv6

Mobile IPv6 wird definiert in ”Mobility Support in IPv6” (Draft 15), weitere Verwendungsfunktionen werden in anderen IETF-Dokumenten z.B im RFC 2461 definiert. Es behandelt folgende drei Hauptelemente:

- Die Funktion Mobile Node (MN): Diese Funktion wird im Mobile Node installiert. Er erfüllt die Funktionen von Bewegungserkennung (Bewegungen von einem Subnetz zu einem anderen) und dem Melden seines gegenwärtigen Standorts gegenüber seinem ”Home Agent” und seinem Korrespondenten betreffend.
- Die Funktion Home-Agent (HA): Diese Funktion wird im Router aktiviert, der das Subnetz mit dem Mobilen Knoten verbindet. Er kümmert sich um den gegenwärtige Standort des Mobilen Gerätes und nimmt Datagramme entgegen um sie an die richtige Stelle (MN) weiterzuleiten.

- Die Bewegungs-Funktion des Correspondent Node (CN): Diese Funktion wird im Correspondent Node aktiviert. Er kümmert sich um den gegenwärtigen Standort des MN betreffend und speichert diese Daten. Die Datagramme werden dann direkt an den MN übertragen.

2.2.1 Allgemeine Beschreibung

Wenn der mobile Node in seinem Home Netz ist, benutzt er die traditionellen Routing Algorithmen um IP-Datagramme mit seinen Korrespondenten auszutauschen. Solange der Mobile Node an sein Home Subnetz angeschlossen ist, verhält es sich wie ein fester Knoten.

1. Der Mobile Node (MN) ist mit seinem Heimnetz verbunden.
2. Der MN öffnet eine Kommunikation mit einem Korrespondentenknoten (CN), bevor er sich in Richtung des fremden Netzes bewegt.
3. Wenn der MN mit einem fremden Netz eine Verbindung aufnimmt, startet er eine Bewegungserkennung z.B. um das neue Subnetz zu erkennen.
4. Der MN erwirbt eine vorübergehende Adresse in dem fremden Netz. Diese Adresse wird Care-of-Adresse (CoA) genannt. die Adressvergabe kann durch eine automatische Konfiguration mit Hilfe einer RA-Nachricht (Router Advertisement) geschehen. Diese RA-Nachrichten werden im Protokoll Neighbor Discovery [RFC-2461] definiert. Das RA wird periodisch vom Router des fremden Sub-Netzwerkes gesendet.
5. Der MN sendet seinen Standort als Nachricht mittels eines Updates (Binding-Update) zu seinem HA und zu seinem Korrespondenten. Das Update ist ein Element, das in den '"Destination Options Header integriert ist. Die Übertragung kann mit aber auch ohne Daten geschehen.
6. Danach werden die Pakete zwischen dem MN und dem CN nur noch direkt zwischen diesen beiden Knoten geroutet. Hier wird kein Tunnel benötigt, jedoch werden die Felder *Source Routing* und *Home Adress* im IPv6-Header gebraucht.
7. Die Pakete die an den MN gesendet werden und beim Heimatnetz des MN eintreffen, werden vom HA entgegengenommen, eingekapselt und an den MN gesendet. Wenn der HA die Originaldaten verpackt, schreibt er einen weiteren IPv6-Header davor und schickt alles zusammen weiter. Als Ursprung dient dann die Adresse des HA und als Zieladresse wird die *care-of-Adresse* des MN eingetragen, die das Ende des Tunnels darstellt.

2.2.2 Detaillierte Beschreibung

Dieses Kapitel beschreibt den Ablauf, wenn ein mobiles Gerät sich durch unterschiedliche IP-Subnetze bewegt.

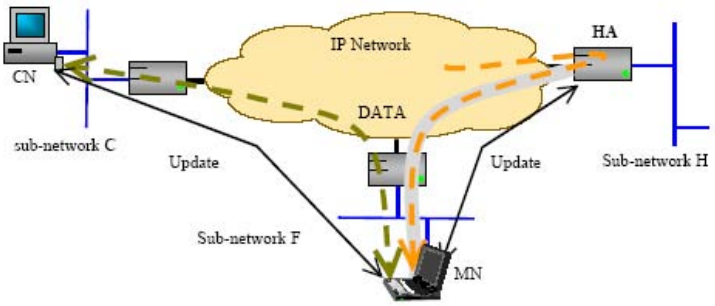

Abbildung 2.1: Mobile IPv6 Überblick

Der mobile Node in seinem Heimatnetz

Im Heimatnetzwerk befindet sich ein Router (HA) der durch ein periodisches Signal (Router Advertisement) seine Anwesenheit anzeigt.

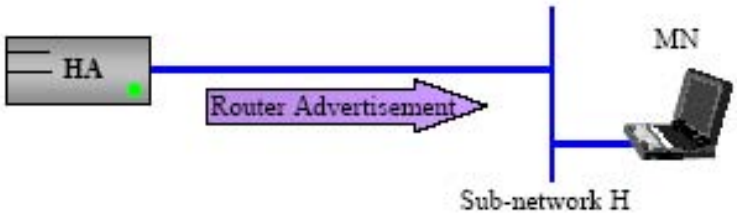

Abbildung 2.2: Übertragung ins Home Netzwerk

Die Router Advertisement (RA) Nachricht ist eine ICMPv6-Nachricht des Neighbor Discovery-Protokolles. Die gegenwärtigen Knoten im Subnetz benutzen diese Nachricht um die Präfixe zu errechnen, die notwendig sind um ein autoconfigure vorzunehmen. Das Mobile-IPv6 hat nun das RA-Format modifiziert und neue Optionen hinzugefügt. Die folgenden Möglichkeiten sind nicht obligatorisch aber ein HA sollte die meisten von ihnen anzeigen können. Mit einem solch modifizierten RA ist ein Knoten fähig um folgende Funktionen ausführen zu können:

- Erkennen aller angeschlossenen HA´s im jeweiligen Sub-Netzwerk,
- Die globale IPv6-Adresse dieser HA´s zu entdecken,
- Erkennen des RA‘s-Intervalls,
- Erkennen des HA und dadurch die Up-Time zu bestimmen.

Damit empfängt ein MN periodisch das RA-Signal seines HA und kann dadurch bestimmen, ob er in seinem Heimnetz befindet. Es muss jedoch Berücksichtigt werden, dass der mobile Knoten eine IP Kommunikation mit einem Korrespondenten öffnet, bevor er sich zu einem fremden Netz bewegt.

Die mobile Node in einem fremden Netzwerk

Der MN trifft ein und nimmt eine Verbindung mit einem Subnetz auf. Die Vorgänge auf Layer 1 und 2 werden hier nicht beschrieben.

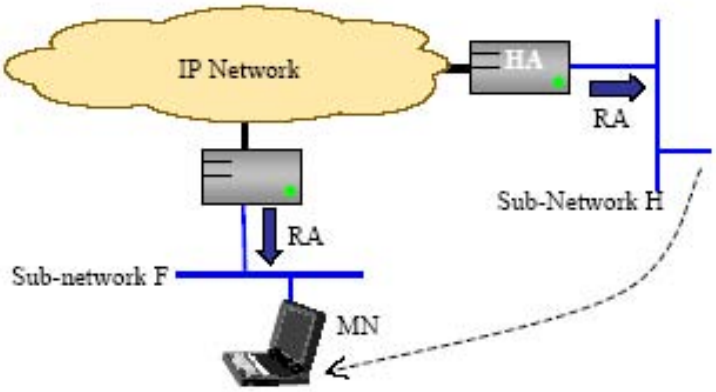

Abbildung 2.3: Verbindung zu einem fremden Netz

Nachdem die Verbindung mit Layer 2 erfolgreich war, erkennt der MN ein neues IP-Subnetz und damit auch, dass er sich in einem neuen IP-Subnetz befindet.

Bewegungserkennung

Um seine Bewegung zu erkennen, benutzt die MN die Neighbor Discovery Protokoll-Mechanismen, wie sie in diesem Kapitel beschrieben werden.

Diese Methode basiert gegenwärtigen auf dem Empfang von ausgesandten RA-Nachrichten in den Subnetzen. Durch das Empfangen von RA-Nachricht aktualisiert und vergleicht der MN seine Präfix- und Routerlisten. Wenn der MN entdeckt, dass er keinen Kontakt mehr zu seinem gegenwärtigen Router hat, wählt er einen anderen aus seiner Routerliste aus und erstellt sich eine IP-Adresse aus dem entsprechenden Präfix-Bereich. Innerhalb des IP-Frameworks ist diese Adresse als Care-of-Adresse gekennzeichnet. Der MN berücksichtigt nun den Kontaktverlust zu seinem Default-Router mithilfe des Neighbor Discovery. Er kann jedoch auch das Advertisment-Interval benutzen, wenn dieses im momentanen Router implementiert und eingeschaltet ist. Nach diesem Intervalablauf wird er meinen, dass es den Kontakt zum Router verloren hat. Im fremden Netz eingetroffen, empfängt der MN kein RA mehr von seinem HA jedoch aber von dem gegenwärtigen Router aus diesem Subnetz. Der MN hält den zuletzt bekannten HA für seinen Default-Router und erstellt nun seine erste Care-of-Adresse (CoA) mit dem neu bekommenen Präfix. Dann meldet sich der MN am ersten HA mit seiner Care-of-Adresse an. Zwar kann der MN mehrere verschiedene Präfixe empfangen und auch verschiedene Care-of-Adressen erstellen, aber nur die erste *primary CoA* wird benutzt um sich bei seinem HA anzumelden.

Mobility Information Update (Binding-Update)

Hier weiß der MN seine Care-of Adresse und es wird berücksichtigt das der MN die Adresse seines HA kennt (danach werden wir uns mit der dynamischen entdecken von HA‘s befassen). Der MN tauscht nun Informationen bezüglich seinen Bewegungen mit dem HA und dem Correspondents aus. Diese Nachrichten werden *Binding Updates* (BU) und *Binding Acknowledgment* (BA) genannt. Diese Nachrichten erlauben dem MN, dem HA und dem Korrespondenten ihre Caches auf dem aktuellen Stand zu halten. (eine Art Liste der Korrespondenten Standorte).

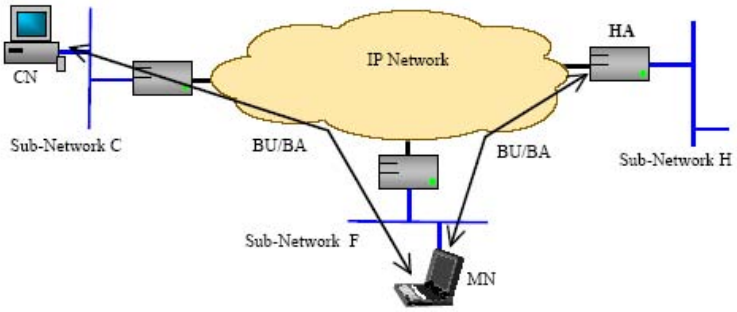

Abbildung 2.4: Binding Updates und Binding Acknowledgment

Binding Update

Die Binding Updates werden vom Mobile Node (MN) zu dem HA und den MN‘s der Korrespontents geschickt. Diese Nachrichten aktualisieren die Standorte der MN‘s und ihre Binding Caches. Tatsächlich ist ein BU eine Option des Destination Option Header welche in IPv6-Datagrammen eingeschlossen werden kann um Daten zu transportieren. Ein BU enthält mindestens die Primäre CoA des MN und die Lebenszeit des Updates (eine Lebenszeit von ”0” bedeutet den Eintrag zu löschen). Einige Sub-Options werden auch definiert um BU mit einem Binding Request zu synchronisiere oder um Alternativen vorzuschlagen sowie CoA zu prüfen. Alle Datagramme die BU mit sich tragen müssen geschützt werden. Im Frühstadium des IETF-Entwurfes nahm man an, dass IPsec der richtige Kandidat für solch einen Schutz war. Jetzt jedoch erscheint IPsec zu komplex und andere Mechanismen werden herrangezogen. Der MN kann einem Korrespondenten ein BU in einer bestimmten Nachricht senden oder warten und das BU dann zusammen mit einigen Daten schicken. Der MN kann BU‘s zu einer bestimmten Liste von Korrespondenten senden oder warten bis die HA‘s das für ihn übernehmen (was aber auch heißt das der Korrespondent nicht alle BU empfängt). Damit kann der MN beschließen seinen Standort dem Korrespondenten nicht preiszugeben.

Ein BU wird gesendet wenn:

- der MN eine Bewegung entdeckt,
- die Lebenszeit des letzten gesandten BU nahezu abgelaufen ist,
- der MN gerade eine Binding Request empfangen hat (Nachricht die geschickt wird um ein neues BU anzufordern).

Binding Acknowledgment - 1

Ein Mobile Node kann von einem Korrespondenten (HA oder andere) eine BU Bestätigung fordern. Danach muss der Korrespondent mit einer Binding Acknowledgment Nachricht antworten. Ein BA bestätigt, dass der Korrespondent sein Binding Cache aktualisiert hat, er kann ihm jedoch auch ein Fehler Code übersenden.

Wie jedes BU muss auch jedes BA geschützt werden.

Binding Acknowledgment - 2

Sobald ein BU akzeptierte wurde, sendet der HA ein Neighbor Advertisment ins Heimnetz. Diese Nachrichten aktualisieren die Caches aller Heimnetz Nodes. Auf diesem Weg verbindet sich die MAC-Adresse des HA mit allen IP-Adressen der MN's unter den Präfixen des Subnetzes. Danach benutzt der HA Neighbor Advertisment um auf Neighbor Solicitation der MN's zu antworten. Diese Mechanismen werden für alle MN Adressen sowie für Adressen in der lokale Umgebung benutzt. Alle zum MN gesendeten Datagramme werden lokal zum HA geschickt. Der HA modifiziert den Routing Prozess und fängt alle Datagramme die zum MN geschickt werden ab um sie einzupacken und zum gegenwärtigen Standort des MN zu schicken.

Binding Acknowledgment - 3

Sobald ein BU akzeptierte wurde, modifiziert der Korrespondent seinen Routing Prozess um die Datagramme zum MN zu senden. Diese Datagramme werden nicht durch das Heimnetz des MN geroutet, sondern sie werden mit einem Routing Headeäusgestattet um durch das gegenwärtige fremde Netz zum MN geroutet.

Datagrams forwarding

Nachdem alle Caches durch Updates auf den neusten Stand gebracht wurden, ist es dem MN möglich die aktive Verbindung zum Korrespondent weiter zu nutzen und auch neue Verbindungen zu öffnen.

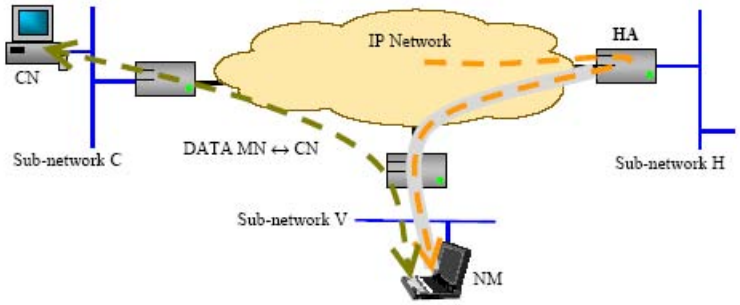

Abbildung 2.5: Datagramm forwarding zwischen dem MN und einem anderen Node

Datagramme MN zu CN

Datagramme die vom MN zum CN gesendet werden, werden direkt vom fremden Netz zum Korrespondentennetz geschickt. Bevor man überhaupt Datagramme versenden kann muss man:

- die Care-of-Adresse des MN als Ursprungsadresse wählen, um Probleme bei der Filterung in Netzen zu umgehen und aus den Netzen heraus zu kommen.
- die Home-Adress hinzufügen so dass der Empfänger die CoA im Source-Adress-Feld durch die Home-Adress des MN ersetzt bevor er auf die höheren Layer wechselt.
- die IP-Adresse des CN als Zieladresse behalten.

Nebenbei erscheinen die Datagramme als ein Teil der gleichen Verbindung zu sein, die im Heim-Netz etabliert wurde. Mobile IP benutzt diese Methode, um Datagramme zu all seinen Korrespondenten einschließlich des Home Agents zu senden. Die Home-Adress Option kann nicht benutzt werden, um den Corresponding Binding Cache zu aktualisieren, außer das Datagramm enthält die BU-Option. Für kurze Nachrichten (DNS-Request, ...) kann der Mobile Node das klassische Forwarding benutzen, ohne Mobile IP-Mechanismen zu verwenden, indem er seine CoA als Ursprungsadresse einsetzt.

Datagramme CN zu MN

Datagramme die vom CN zum MN gesendet werden, werden direkt vom fremden Netz zum Korrespondentennetz geschickt. Bevor man überhaupt Datagramme versenden kann muss man:

- die Care-of-Adresse des MN als Zieladresse wählen, um die Datagramme direkt durch das fremde Netz dem Mobile Node zu schicken und damit die Versendung durch den HA zu vermeiden.
- die Routing Header Option hinzufügen, die die Home Adress des MN einschließt, so dass der MN die CoA der Zieladresse durch die Home Adress des MN ersetzt, bevor die Daten zu höheren Layern übergeben werden.
- die IP-Adresse des CN als Ursprungsadresse zu behalten.

Nebenbei erscheinen die Datagramme als ein Teil der gleichen Verbindung zu sein, die im Heim-Netz etabliert wurde. Alle Correspondents einschließlich der HA verwenden diese Methode um Datagramme zum Mobile Nodes zu senden.

Intercepted datagramme HA zu MN

Der Correspondent sendet die zum Mobile Node gehörenden Datagramme in das Heimnetz des MN so lange, wie er kein Binding-Update bezüglich der Bewegung des MN bekommt. Diese Datagramme werden vom HA aufgehalten, in ein anderes IPv6-Datagramm eingekapselt und dem Mobile Node im fremden Netz geschickt. Der Encapsulation Header hat als Zieladresse die Care-of-Adresse des MN und die IP-Adresse des HA als Ursprungsadresse. Um das ursprüngliche Datagram nicht zu modifizieren, benutzt der HA den Encapsulation Header. Für Datagramme vom Sender verhält man sich wie alle anderen Korrespondenten und benutzt den Routing Header anstelle des Encapsulation Header. Wenn man ein eingekapseltes Datagramm empfängt sendet man ein Binding-Update zum Ursprung um die Verbindung über den HA zu umgehen. Der Mobile Node kann sich auch dafür entscheiden alle Datagramme über den HA zu leiten und damit seinen Standort geheim zu halten.

Multicast Datagrams

Die IETF diskutiert noch wie man IP Multicast bei der Benutzung Mobile IP optimieren kann.

Zurück zum Home-Netz

Das Mobile Node erkennt seine Bewegungen und seine Rückkehr auf das Home-Subnetz dank des RA von seinem HA. Der MN registriert sich wieder bei seinem HA und teilt ihm mit das er wieder unter seiner Home Adress erreichbar ist und der HA alle noch bestehenden verweise löschen kann. Der HA antwortet mit einem Binding Acknowledgment. Nach diesem Acknowledgment hört der HA auf sich als Proxy für den MN zu verhalten. Danach muss natürlich vom MN ein Neighbor Advertisment geschickt werden um die IP Adresse mit der MAC Adresse auf Layer 2 zu verbinden. Wenn man dies nicht macht würden die Datagramme aus dem Home Netz weiterhin zum HA geleitet werden, bis der Timeout des Caches erfolgt und die Tabellen sowieso neu geschrieben werden. Am Ende dieser De-registration benutzt der HA und der MN wieder die klassischen Verbindungsmethoden für die Datenpackete. Der HA beseitigt danach die Informationen mit denen man ein Bewegungsprofil eines Benutzers erstellen könnte. Dies geschieht durch das senden des BU mit seiner Heim-Adresse als Ursprungsadresse.

2.2.3 Weitere Methoden

Mobility agent solicitation

Für das Router Discovery werden zwei Nachrichten benötigt. Die Router-Advertisement Nachrichten werden von den Routern periodisch an alle anderen Knoten versendet. Darin übermitteln sie Informationen über ihre Adresse, ihre Lebensdauer, usw. Für Mobile IPv6 hat man noch ein Bit hinzugefügt. Mit diesem Bit kann der Router angeben, ob er auch als Home Agent fungieren kann. Ein Router kann auch außerhalb dieser Periode eine Router-Advertisement Nachricht an einen einzelnen Host schicken. Dies geschieht, wenn der Host ihm eine Router Solicitation-Message schickt.

Dynamic Home Agent Discovery

Der Mobile Node darf die IP-Adresse seines Home Agents nicht kennen (weil diese Adresse nicht auf dem MN konfiguriert worden ist). Der MN wird versuchen die IP-Adresse seines HA dynamisch zu entdecken, wenn er sich in einem fremden Netz befindet. Der MN sendet eine Home-Agent-Adress-Discovery-request genannte ICMP-Nachricht zu seinem Home Netz. Sobald ein HA diese Nachricht erhält, antwortet der HA mit einer Home-Agent-Adress-Discovery-reply genannten ICMP-Nachricht. Diese Nachricht schließt die Liste der IP-Adressen aller gegenwärtigen HAs auf dem Home Netz in bevorzugter Reihenfolge ein. Der MN kann nun wählen zu welchem HA er sich verbinden will. Jeder HA hat dank der RA-Nachrichten eine Liste von allen HA zur Verfügung.

Binding Request

Ein Correspondent Node, der in Verbindung mit einem Mobile Node ist und seinen Eintrag im Cache des Correspondent Node erneuern will, kann ein Binding Update schicken, sofern er noch keinen BU bekommen hat. Dieser Request ist eine Option die in ein Packet von Daten eingepackt werden kann. Nach dem Empfang dieses Requests kommt als Antworten ein Binding Update, um z.B. seinen neuen Standort in den Cache des Correspondent Node zu schreiben. Wenn der Mobile Node sein Standort nicht enthüllen will, antwortet es mit einem BU, bei dem er die Lebenszeit auf null setzt und er als Care-of-Adresse seine Home Adress einsetzt.

Traffic Forwarding

Wenn der Mobile Node sich mit einem fremden Netz verbindet, nachdem er eins verlassen hat, muss er seinen HA und seinen Korrespondent den neuen Standort mitteilen. Dieses Update erlaubt dem HA und dem Korrespondenten seine Datagramme nun in das neue fremde Netz zu senden. Andererseits treffen alle vor diesem Update gesandten Datagramme in dem alten fremden Netz ein. Wenn ein HA auf dem alten fremden Netz vorhanden war, kann der MN fragen, ob der HA ein Traffic Forwarding macht, um die Datagramme sofort in das neue Netz zu senden. Zuerst muss der MN aber ein BU zu dem HA senden. Diese BU-Nachricht muss die neue CoA als Ursprungsadresse (die der MN in dem neuen Unternetz erworben hat) und als Heim-Adresse die CoA vom alten fremden Netz enthalten. Das Mobile Netz erkennt die Anwesenheit eines HA in einem Subnetz durch das RA des Subnetzes oder durch den HA-Discovery Mechanismus.

Diese *Binding Updates* müssen auch geschützt werden.

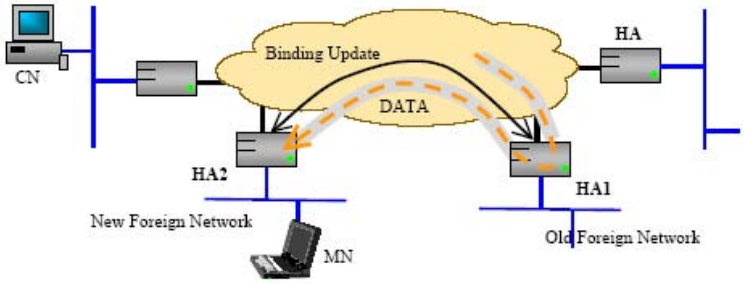

Abbildung 2.6: Traffic zwischen HA1 und HA2

2.2.4 Abschluss

Mobiles IP bringt zwei der mächtigsten Technologietrends der Welt zusammen, dem Internet und der mobilen Kommunikation. Die Mobile IP Lösung erlaubt einem Mobile Node jede Verbindung zu halten, auch wenn er das Subnetz wechselt. Der Mobile Node ist unter seiner Home Adress erreichbar, egal in welchem Subnetz er sich gerade befindet. Obwohl Mobile IP ein ziemlich gutes Konzept ist, gibt es doch noch einiges was untersucht werden muss. Die gegenwärtige Forschung in Mobile IP konzentriert sich auf Gesprächsthemen wie verbesserte Sicherheitsdienstleistungen und verbessertem QoS. Zum

Beispiel ist die Art und Weise, Updatenachricht zu sichern (Binding-Update, Request und Acknowledgment) bei IETF noch nicht *stable*. Zudem gibt es noch viel zu tun im Bereich des Fast Handover, dem Verlust von Packeten und der Unterbrechungszeit wenn ein MN eine Bewegung entdecken und sein Korrespondenten seinen Cache aktualisieren muss.

2.3 Mobile IPv4 im Vergleich zu Mobile IPv6

Mobile IPv4 hat viele Features die auch bei Mobile IPv6 vorzufinden sind. Bei der Entwicklung von Mobile IPv6 hat man die Schwächen von Mobile IPv4 beruecksichtigt. Die Hauptunterschiede zwischen Mobile IPv4 und Mobile IPv6 sind:

- Unterstützung von „route optimization“. Dieses Feature ist jetzt als ein fundamentaler Teil von Mobile IPv6 in das Protokoll eingebaut.
- Bei der Benutzung von CoA als Quelladresse in dem IP-Header des Pakets wird das Routing von „multicast packets“, die von MN abgesendet sind, einfacher.
- In Mobile IPv6 kann die Funktionalität des FA durch die bei IPv6 vorhandenen Features wie “Neighbour Discovery„ und „Address Autoconfiguration“ ersetzt werden. Deswegen wird in Mobile IPv6 auf FA verzichtet.
- Mobile IPv6, anders als Mobile IPv4, benutzt IPSec für alle Sicherheitsanforderungen. In Mobile IPv4 sind Sicherheitsanforderungen durch eigene einzelne Sicherheitsmechanismen für jede Funktion realisiert.
- Mobile IPv6 benutzt „source routing feature“ Dieses Feature macht es für CN möglich, Pakete zu MN zu senden. Wenn MN nicht in „home network“ ist, wird ein IPv6 „routing header“ statt IP-Kapselung benutzt, während Mobile IPv4 die Kapselung für alle Pakete einsetzen muss. Trotzdem ist es in Mobile IPv6 für HA erlaubt, die Kapselung für Tunneling zu nutzen.
- In Mobile IPv6 werden die Pakete, die „home network“ erreichen und für MN, der jetzt nicht in „home network“ ist, bestimmt sind, bei MNs HA unter Benutzung von IPv6 „Neighbour Discovery“, nicht ARP wie bei Mobile IPv4, abgefangen.

Konzept Mobile IPv4	**Konzept Mobile IPv6**
Terminologie, die in beiden Konzepten verwendet wird: Mobiler Knoten, Heimatagent, Heimatnetz, Fremdnetz	
Heimatadresse des mobilen Knotens	Global routbare Heimatadresse und link-local Heimatadresse
Fremdagent	Nur ein IPv6 Router im fremden Netz (Fremdagenten existieren nicht mehr)
Care-of Adresse beim Fremdagenten	Es gibt nur noch co-located Care-of-Adressen
Co-located Care-of-Adresse	
Care-of-Adresse wird via Agent Discovery, DHCP oder manuell vergeben	Care-of-Adresse wird via Stateless Address Autoconfiguration vergeben
Agent Discovery	Router Discovery
Authentifizierte Registrierung mit dem Heimatagenten	Authentifizierte Notification mit Heimatagent und anderen Kommunikationspartnern
Routing zu mobilen Knoten via Tunnel	Routing zu mobilen Knoten via tunneling und Source-Routing
Routen Optimierung via separater Protokoll-Spezifikation	Routen Optimierung wird unterstützt

Abbildung 2.7: Mobile IPv6 im Vergleich zu Mobile IPv4

Kapitel 3
Realisierung

3.1 Konzept

Um eine Grundlage für die Testläufe zu haben, musste im Vorfeld eine Konzeptionierung erfolgen anhand derer die Testläufe durchgeführt werden können

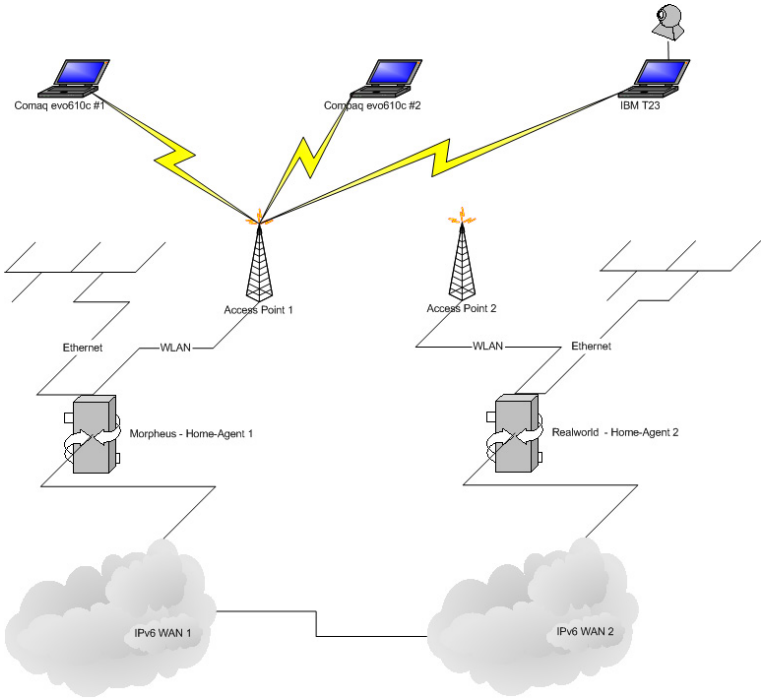

Abbildung 3.1: Grober Aufbau des verwendeten Netzes.

3.1.1 Allgemeines Konzept

Wie in Abbildung 3.1 zu sehen ist wurde versucht an 2 unabhängigen Netzen zu arbeiten. Beide Gateways (**realworld** & **morpheus**) haben jeweils 2 unabhängige IPv6 Netze mit einer Prefixlänge von 64BIT fuer EUI64[1]. Jedes Gateway besitzt 3 Interfaces. Inter-

[1]EUI64 definiert die auto-address Configuration. **RFC 2464**

nes Ethernet, zur Anbindung der Notebooks, WLAN-Interface, an dem der Access-Point angebunden ist und ein Externes Interface zur Anbindung an das LAN im Wohnheim.

Die IPv6 Netze werden über einen IPv6 in IPv4 Tunnel zu den Gateways bereitgestellt. 2 der Netze wurden freundlicherweise von der Universität Erlangen bereitgestellt, die 2 weiteren Netze wurden durch Tunnelbroker realisiert.

Dies ermöglichte es Mobile IPv6 Tests durchzuführen bei der die Kommunikation über das WAN erfolgt. Durch die erhöhte RTT könnten Probleme beim verschicken der “binding update” Nachrichten enstehen, die man bei Tests im LAN nicht erkennen kann.

Die Software, die wir fuer die Testszenarios benutzt haben waren zum einen das nicht bandbreiten- und zeitkritische Programm SSH (Secure Shell), zum anderen sollte aber auch eine Anwendung getestet werden, die viel Bandbreite nutzt um zu testen, wie sie auf ein Handover zu einem anderen Netz reagiert.

Für den bandbreiten intensiven Test entschieden wir uns fuer die CVS Version von GnomeMeeting. GnomeMeeting ist eine H.323 kompatibles Videokonferenz und VOIP/IP Telefon Applikation. Mithilfe dieser Applikation ist es möglich einen praxisnahen Test von UMTS ähnlichen Diensten zu testen.

Vorraussetzung

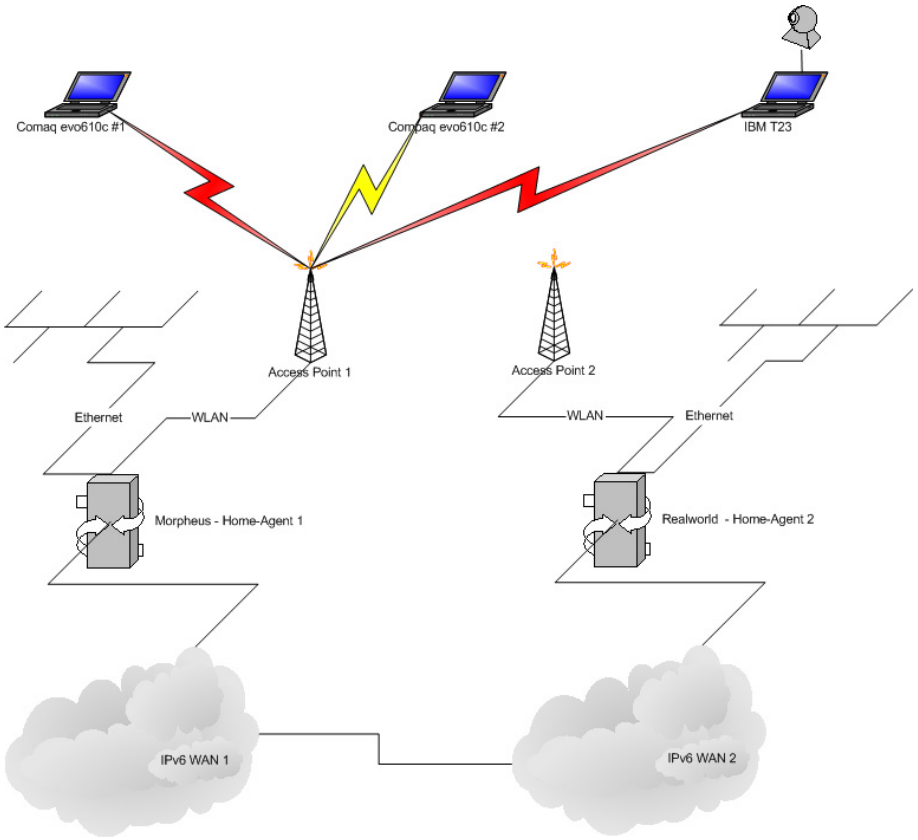

Abbildung 3.2: Netzwerkaufbau Vorraussetzungen.

Als Grundvorraussetzung sollten alle Notebooks via WLAN am Router **morpheus** angebunden sein. Es wird (entweder mit ssh oder mit gnomemeeting) eine Verbindung zwischen 2 Notebooks hergestellt. Alle Notebooks haben eine HomeAddress die auf dem WLAN Interface von Morpheus liegt. Somit ist die Verbindung zwischen 2 Notebooks link-local.

Hierbei sollten keine grösseren Probleme auftauchen da jeder IPv6 Stack einen normalen Connect aufbaut und noch keine Binding Updates verschickt werden müssen.

Der Mobile IPv6 Stack muss lediglich seine HomeAddresse als Source-IPv6 verwenden. Da durch die stateless Autoconfiguration der Mobilenode sowohl seine HomaAddress alsauch eine Care-Of-Address erhält, muš der Mobile IPv6 Stack dafür sorgen das Connections nur von der HomeAddress aus gehen.

WLAN-Wechsel

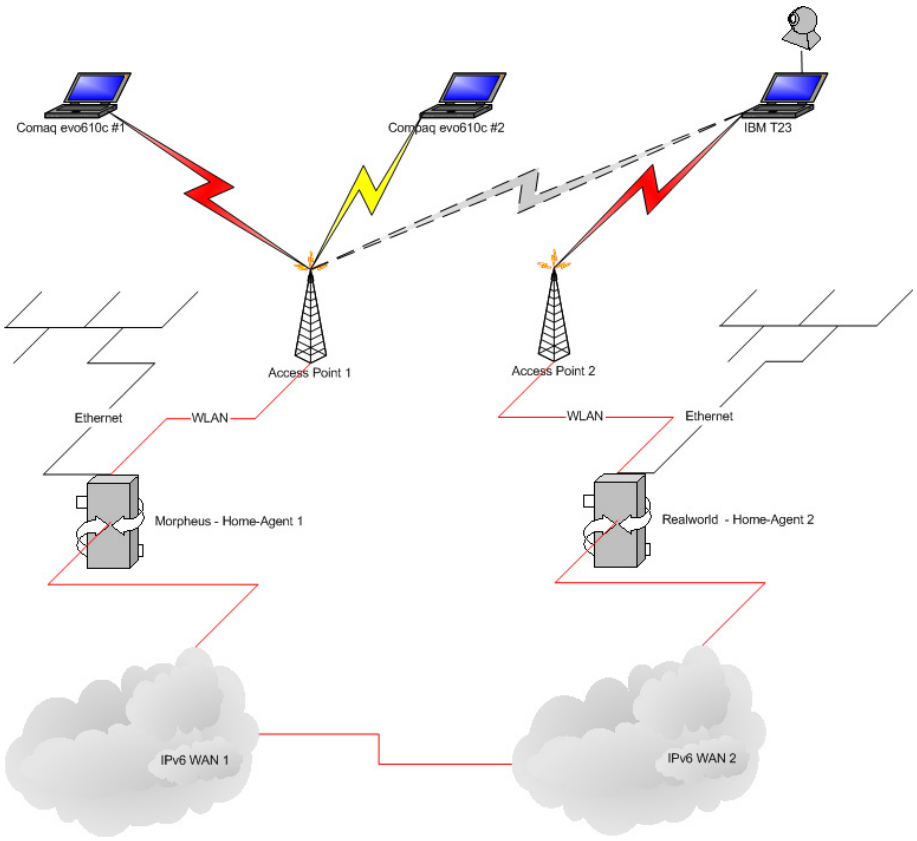

Abbildung 3.3: Wechsel des WLAN Netzes.

Wenn nun eines der Notebooks zu dem anderem WLAN Netz wechselt wird sich der Netzverkehr bereits über das WAN hinaus begeben. Hierbei gilt dann **realworld** als Foreign Agent. Bei diesem Wechsel muss dann das wechselnde Notebook einen Binding Update an **morpheus** schicken. Bestehende Verbindungen werden durch anfüegen von Destination Option Headern benachrichtigt das die HomeAddress nun über die neue Care-of-Address erreichbar ist. Dies sorgt für eine reibungslose Verbindung trotz Netz Wechsel.

Aufgrund der fehlenden Fast Handover Implementation ist dieses Szenario sehr Fehleranfällig. Es fehlt hierfür eine einfache Möglichkeit für den Mobile IPv6 Stack zu überprüfen ob ein WLAN Wechsel bevorsteht. Dadurch könnte das System sich auf den Wechsel vorbereiten und im besten Falle automatisch das WLAN Netz mit der besten Empfangsqualität aneignen.

In Anbetracht der im Vorfeld bekannten Probleme haben wir uns noch für einen weiteren Wechsel entschieden, bei dem nicht innerhalb eines Interfaces der Wechsel stattfindet sondern das Interface selber gewechselt wird.

Interface Wechsel

Beim Interface Wechsel wird der HomeAgent mit 2 Aufgaben betraut. Am Ethernet Interface wird er zum Foreign-Agent für die Notebooks und empfängt an die IPv6 Addresse

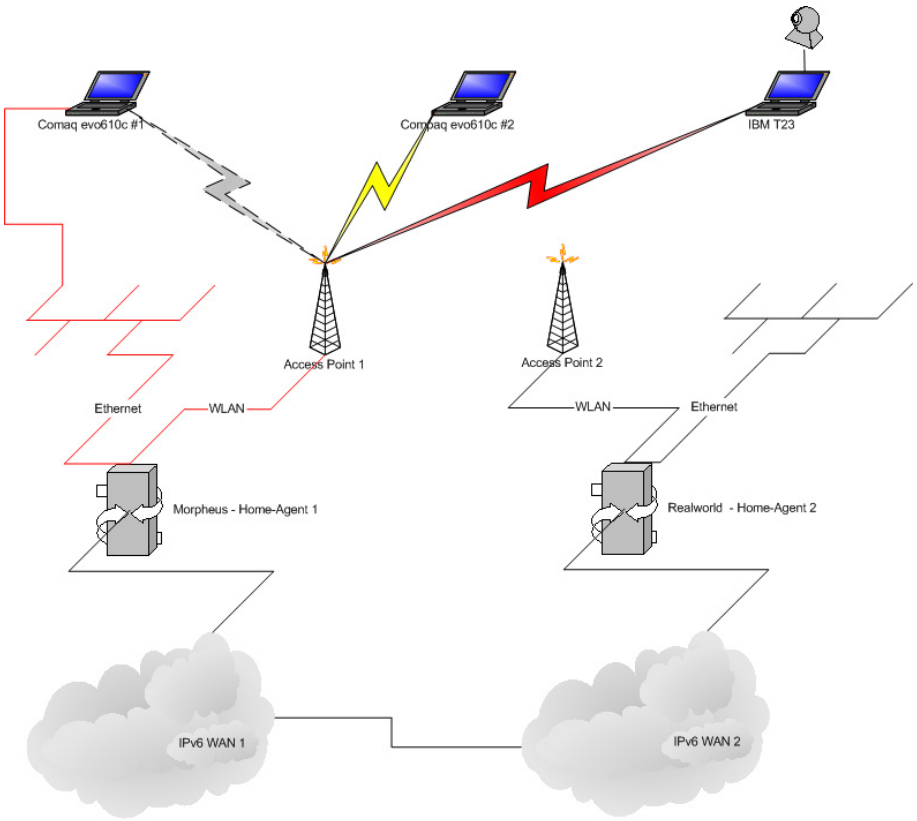

Abbildung 3.4: Wechsel des Interfaces an einem Notebook.

am WLAN Interface die HomeAgent spezifischen Updates. Dieser Wechsel wird dadurch erreicht das am Notebook das WLAN Interface mittels einem `ifconfig wlan0 down` und einem `ifconfig eth0 up` das Hauptinterface sich wechselt.

Hierbei erhält das Notebook eine Care-of-Address am 2. Interface und schickt ein Binding Update an den HomeAgent.

Tests mit einem Corresponding Node

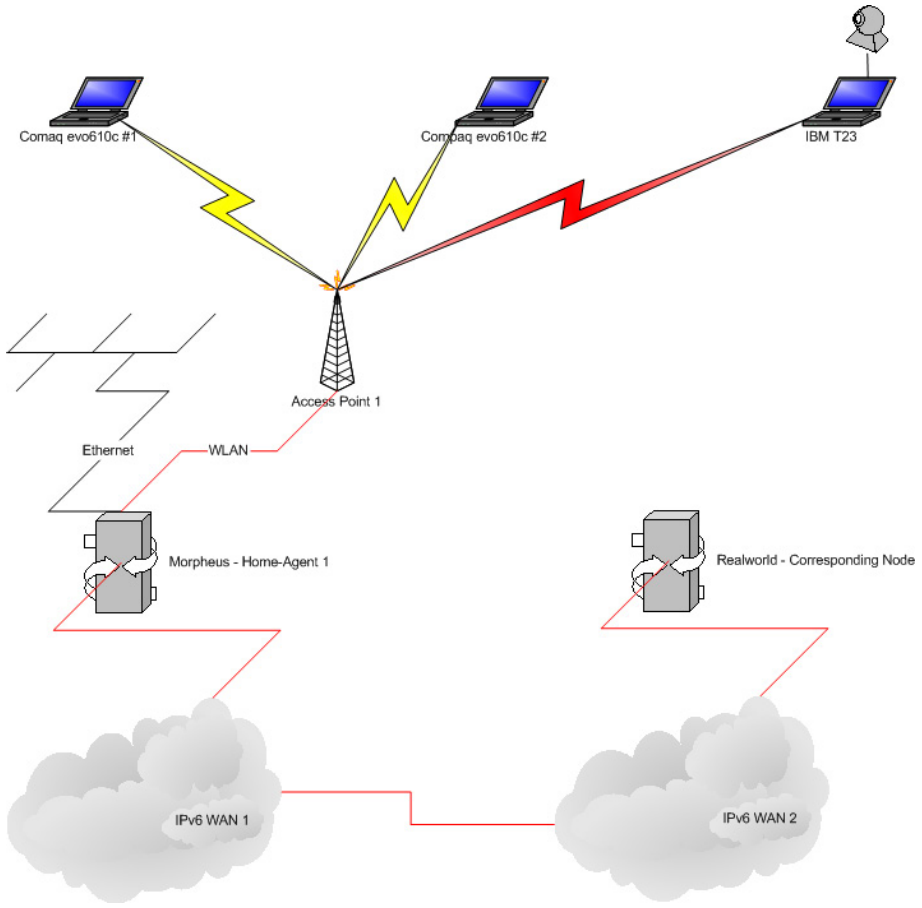

Abbildung 3.5: Traffic zwischen Mobilei Node und Corresponding Node ohne Mobile-Node Features

Damit nicht nur Tests mit Mobile Nodes stattfinden wurde auch ein Corresponding Node und eine Machine eingebunden die keine Mobile IPv6 Fähigkeiten besitzt. Hier wurde dann **realworld** als CN ausgesucht, der Aufgrund der Inkompatibilität nicht als Home-Agent für linux Mobile Nodes agieren kann.

Bei einer Verbindung zwischen einem Mobile-Node und einem IPv6 Stack ohne Mobile IPv6 Unterstützung sollte der HomeAgent einen Fallback Tunnel aufbauen zum Mobile-Node. Über diesen wird dann Traffic von und zu dem Mobile Node, der von nicht Mobile-IPv6 fähigen Machinen kommt, geroutet.

Als geeigneten Test hierfür wurde ssh verwendet. Da der Router **realworld** keine X11 Oberfläche besitzt wurde Gnomemeeting nicht weiter berücksichtigt.

Wechsel des Interfaces

Beim Wechsel des Netzes wird der MobileNode an den CN eine Binding Update Nachricht mitschicken.

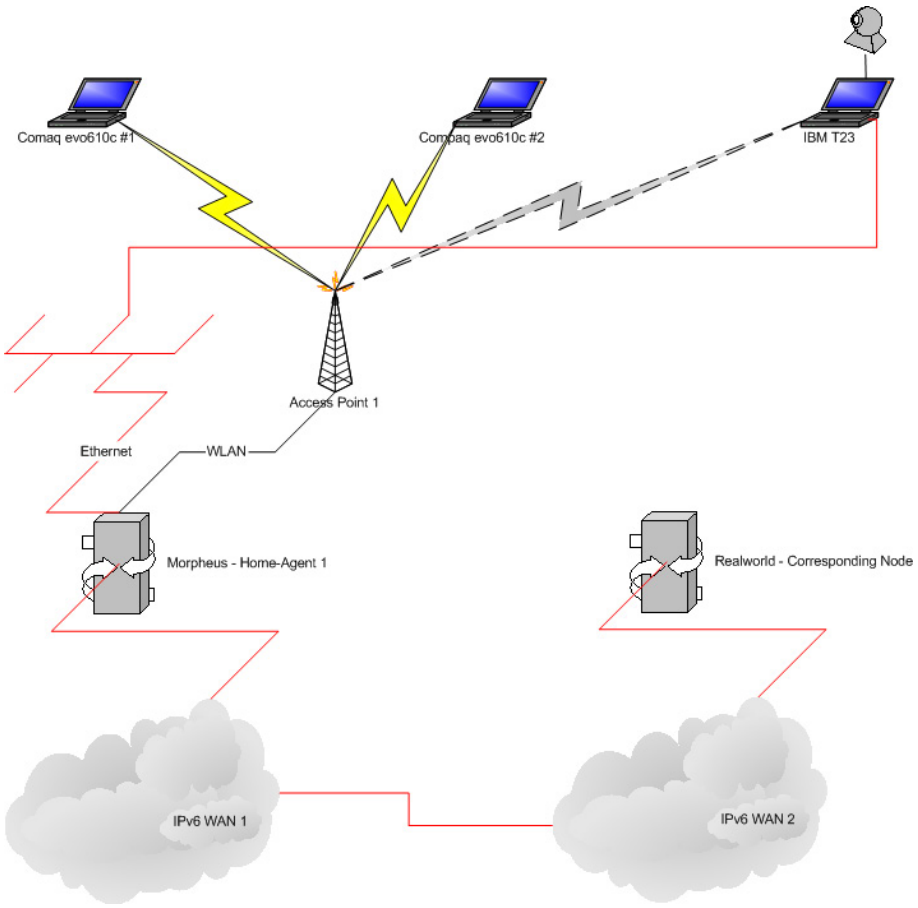

Abbildung 3.6: Wechsel des Netzes bei Kommunikation mit Corresponding Node

3.1.2 Test Konzept

Nach ein paar wenigen Versuchen stellte man leider fest, dass das Konzept leider nicht aufgeht, denn der FreeBSD Router, der als Home Agent eingesetzt werden sollte, wollte mit den Linux basierten Mobile Nodes nicht zusammenarbeiten.

Das Konzept wurde überarbeitet. Jetzt sollten die Anwendungen einem simulierten WLAN-Wechsel standhalten. Der Wechsel wurde simuliert, indem man statt über einen zweiten Access-Point sich in ein fremdes Netz per WLAN einbuchte, dies über einen simplen Interfacewechsel vollzog. Die Zeitabstände, in denen sich der Router im Netz advertisen sollte war auf 1,5 Sekunden eingestellt.

Beim simuliertem Wechsel entstanden die in Kapitel 4.3 beschriebenen Probleme. Es war nicht möglich die von Hewlett-Packard gestellten Notebooks für diese Tests zu benutzen. Daher wurde das IBM T23 verwendet um diese Interface Wechsel durchzuführen.

3.2 Analyse der verwendeten Software

In den folgenden Abschnitten wird die Software erklärt die wir verwendet haben. Es wird genau aufgelistet wie man schritt für schritt die Software installiert. Natürlich konnten wir hier nicht erkären wie man einen kompletten Kernel kompiliert, dafür gibt es im

Internet genügen Anleitungen. Es wird in den meisten Fällen nur der spezielle Teil beschrieben der direkt mit unserem Projekt zu tun hat.

Nach einer gewissen Einarbetungszeit blieben als Betriebssysteme nur noch 2 Kandidaten übrig. Wir hätten gerne ein Windows mit benutzt, was daran scheitern musste das Microsoft die Entwicklung eingestellt hat.

Auf den Notebooks haben wir uns, aufgrund des besseren Hardware Supports, für ein Linux entschieden. Als HomeAgents wollten wir sowohl ein Linux als auch ein FreeBSD verwenden.

3.2.1 USAGI-Projekt

Allgemeines

USAGI[2] steht für "UniverSAl plAyground for Ipv6" und ist ein Projekt, daß sich ausschliesslich mit der Entwicklung vom IPv6-Stack und IPSec (für IPv4 und für IPv6) für das Linux-System beschäftigt. Es arbeitet eng mit dem WIDE-, KAME- und TAHI-Projekt zusammen. Das Projekt wird von Freiwilligen Entwicklern unterschiedlicher japanischer Organisationen unterstützt.

Alle zwei Wochen bringt das USAGI-Projekt einen Snapshot, sozusagen eine Entwicklerversion, heraus, die nicht in einer produktiven Umgebung verwendet werden sollte. Darueber hinaus gibt es mehrmals im Jahr eine Stable Release. Die aktuellste ist vom 14. Februar 2003 und ist in der Version 4.1 verfügbar.

Für unser Projekt wurde anfangs die Stable Release verwendet, doch angesichts der in Kapitel 4 Abschnitt 4.2 beschriebenen Probleme, probierten wir natuerlich auch den zur Zeit des Testverfahrens aktuellen Snapshot aus.

[2] http://www.linux-ipv6.org

Kompilierung und Installation von USAGI MIP6

- Kernel Kompilierung

  Schon während der Kernel Konfiguration muss entschieden werden, ob die Maschine, auf der der Kernel laufen soll, eine Mobile Node (MN) oder ein Home Agent (HA) ist oder einfach nur als Correspondent Node (CN) agiert. Die empfohlene Kernel Konfiguration sieht dann wie folgt aus (hier wird vorausgesetzt, dass der Leser mit der Kernel Kompilierung und installation vertraut ist):

```
Code maturity level options -->
        [*] Prompt for development and/or incomplete code/drivers
Loadable module support -->
        [*] Enable loadable module support
General setup -->
        [*] Networking support
        ...
        [*] Sysctl support
Networking options  --->
        ...
        <*/M> Unix domain sockets
        [*] TCP/IP networking
        ...
        [*] Network packet filtering (replaces ipchains)
        ...
        <*/M>  IP: tunneling
        ...
        <*/M>  The IPv6 protocol (EXPERIMENTAL)
        ...
        <M>      IPv6: IPv6 over IPv6 Tunneling (EXPERIMENTAL)
        ...
        <M>      IPv6: Mobility Support (Correspondent Node)
File systems  --->
        [*] /proc file system support
```

  Desweiteren müssen folgende Optionen für einen MN Kernel eingestellt werden:

```
Networking options  --->
        [*]      IPv6: sub-tree in routing table support (EXPERIMENTAL)
        ...
        [*]      MIPv6: Mobile Node Support
```

  Genauso für einen HA:

```
Networking options  --->
        [*]      IPv6: anycast support
        ...
        [*]      MIPv6: Home Agent Support
```

- Usagi user-land Installation

  Der Pfad für die user-land Anwedungen ist auf `/usr/local/v6` voreingestellt. Die folgenden Anwendungen findet man also unter `/usr/local/v6/sbin`.

  Mipdiag: diese Anwendung erlaubt es genaue Diagnosen über den Verbindungszustand vorzunehmen.

```
% cd $USAGI_TOP/usagi/mipdiag
% ./configure
% make

% su
# make install
```

Folgend muss noch ein device (Garät) angelegt werden.

```
# mknod /dev/mipv6_dev c 0xf9 0
```

Radvd: das ist der Router Advertisment Deamon für den HA, der MIP6 unterstützt.

```
% cd $USAGI_TOP/usagi/radvd
% ./configure
% make

% su
# make install
```

- Konfiguration

  Das Konfigurationsfile findet man unter `/usr/local/v6/etc/network-mip6.conf`.

  Für eine CN sollte folgende Zeile enthalten sein:

  ```
  FUNCTIONALITY=cn
  ```

  Folgene Zeilen für den Betrieb einer MN:

  ```
  FUNCTIONALITY=mn
  HOMEDEV=eth0
  HOMEADDRESS=3ffe:ffff:1234:5678::abcd/64
  ```

  Und letztendlich die für einen HA:

  ```
  FUNCTIONALITY=ha
  HOMELINK=eth1
  ```

  Um das Router Advertisment nutzen zu können, muss der radvd konfiguriert und gestartet weden. Das Konfigurationsfile findet man unter `/usr/local/v6/etc/radvd.conf`.

  ```
  interface eth1 {
          AdvSendAdvert on;
          AdvHomeAgentFlag on;
          AdvHomeAgentInfo on;

          prefix 3ffe:ffff:1234:5678::1111/64 {
                  AdnOnLink on;
                  AdvAutonomous on;
                  AdvRouterAddr on;
          };
  };
  ```

- Einsatz von USAGI

  Ein Skript zum starten von Mobile IP6 befindet sich nach der Installation unter `/usr/local/v6/etc/init.d/mobile-ip6`.

  Das Starten einer CN oder einer MN geschieht auf gleiche Weise mit dem Befehl:

  ```
  # /usr/local/v6/etc/init.d/mobile-ip6 start
  ```

  Zum Starten eines HA muss man vor der Ausführung von Mobile IP6 zusätzlich den Router Advertizer starten:

  ```
  # /usr/local/v6/sbin/radvd -C /usr/local/v6/etc/radvd.conf
  # /usr/local/v6/etc/init.d/mobile-ip6 start
  ```

Wenn alle Dienste ordnungsgemäss gestartet wurden, kann man den Status der Verbindung mittels `mipdiag` überprüfen. Der HA sollte sich nun im Netz ankündigen und den MN's eine Mobile IP zuweisen.

3.2.2 Installation der Compaq USW WLAN Karte

Die eingebaute USB-WLAN Karte in den Compaq Notbooks werden von den verwendeten Kernels zu Zeit noch nicht direkt unterstützt. Deshalb mussten besondere Treiber verwendet werden die dann als Module Kompiliert werden damit die mit dem Kernel zusammenarbeiten. Zu Beginn des Projektes waren die Treiber[3] der Version 0.1.4 aktuell aber zum jetztigen Zeitpunkt gibt es schon neuere Versionen die schon in den CVS-Tree des neuen Entwicklungs Kernels übernommen wurden.

Die Installtaion der Treiber gestaltet sich relativ einfach. Es müssen nur bestimmte Vorrausetzungen erfüllt sein damit dies einfach von statten gehen kann. Die Orinoco USB Treiber werden aus der Firmware von Orinoco und aus Komponenten aus dem Kernel Source Tree gebaut. Deshalb muss beim Kompileren der Treiber bereits eine Verbindung mit dem Internet bestehen und man muss entpackte Kernel Sourcen auf dem Rechner liegen haben.

```
Software needed for firmware download:
--------------------------------------

                * wget  (http://www.gnu.org/software/wget/wget.html)
                * curl  (http://curl.haxx.se/)
                * unzip (http://www.info-zip.org/pub/infozip/UnZip.html)

        cURL is used to make firmware's download faster, getting
        only the bits the driver needs.
```

Die Treiber verfügen über eine Installtionsscript die es einem sehr leicht machen die Installtion durchzuführen. Man muss nur nach dem Anstossen des Scripts den Anweisungen auf dem Bildschirm folgen. Dies sollte jedem Möglich sein und deshalb werden wir uns hier nur auf das Anschubsen des Scriptes beschränken.

```
        This script allow building, installing and upgrading the
        Linux drivers for Lucent/Agere Orinoco USB devices.

         $ cd /directory/of/drivers
         $ su
         # ./util/INSTALL_ORiNOCO_USB.sh
```

Nach Erfolgreicher Kompilierung der Treiber entstehen 3 Module die in einer bestimmten reihenfolge geladen werden müssen. Es liegt die Überlegung nahe die Module in die */etc/modules* von Linux einzutragen ddamit sie beim Reboot automatisch geladen werden. Leider bringt Linux die Reihenfolge durcheinander wenn man sie beim Starten Automatisch laden lassen will. Deshalb sollten die Module per Hand oder durch ein selbst geschriebenes Start Script geladen werden:

```
        # insmod driver/hermes.o
        # insmod driver/orinoco.o
        # insmod driver/orinoco_usb.o
```

[3] http://orinoco-usb.alioth.debian.org/

Nach dem Laden der Treiber steht nun nichts mehr dem Benutzen der WLAN Karte im Weg.

3.2.3 KAME Projekt

Allgemeines

KAME ist ein joint effort von 7 japanischen Firmen das es sich zum Ziel gesetzt hat einen Referenz IPv6/IPSec Stack zu entwickeln. Durch dieses Projekt soll vermieden werden das unnötige doppelte Entwicklungen in diesem Bereich stattfinden. Das Hauptziel ist die Referenzimplementation auf BSD Systemen bereitzustellen. Derzeit wird der KAME Ipv6 Stack in Free-/Net- und OpenBSD verwendet.

Das KAME Projekt stellt SNAPKITS mit den aktuellen Entwicklungen zur Verfügung in denen u.A. auch die von uns benötigte Mobile IPv6 Funkionalität enthalten ist.

Fr̈ unser Projekt wurde das SNAPKIT vom 5. Mai 2003 für FreeBSD 4.8-STABLE verwendet. Das Snapkit besteht einmal aus einem Kernel-Patch und zu anderem aus sog. Userland-Tools.

Kompilierung und Installation von KAME MIP6

- Kernel Kompilierung

  Im Gegensatz zum USAGI Projekt ist es beim Kernel compilen nicht notwendig zu wissen ob die Machine als Mobile Node oder als Home Agent eingesetzt wird. Daher weicht der standard Kernel nur durch 2 Zeilen vom Mobile IPv6 Fähigen Kernel ab:

  ```
  options "MIP6"
  options "MIP6_DEBUG"
  ```

  Damit kann dann mittels

  ```
  /usr/sbin/config CONFIGFILE
  cd ../../compile/CONFIGFILE
  make depend all install
  ```

  der Kernel kompiliert und installiert werden.

- Userland Installation

  Die Userland Applikationen, die zur Konfiguration benötigt werden, lassen sich mittels einem `make includes includes-install all install` übersetzen. Um ein bestehendes FreeBSD nicht durch die Entwickler-Tools zu beschädigen werden die KAME Applikationen nur in `/usr/local/v6` installiert.

3.2.4 SSH - Secure Shell

SSH[4] ist die Abkürzung für Secure Shell. Anders als Telnet wir bei SSH eine verschlüsselte Verbindung aufgebaut, somit wird der Benutzername sowie das Passwort verschlüsselt übertragen. Mittels eines SSH-Tunnels lassen sich alle möglichen Verbindungen, wie

[4] http://www.openssh.com/

z.B. ftp, pop usw. verschlüsselt aufbauen. Die Funktionsweise von SSH ist nicht Teil dieses Dokuments und wir somit nicht genauer aufgeführt.

Um den Handover bei Mobile IPv6 zu testen, wurde unter anderem auch SSH verwendet, um zu verdeutlichen, wie es auf einen Wechsel der Route reagiert.

Mittels eines SSH-Clients wurde beim Test eine Verbindung zu unserer CN, **realworld**, aufebaut. Erwartungsgemäß sollte die Verbindung auch nach dem Interface-Wechsel bestehen bleiben. Da SSH eine Bandbreiten unintensive und nicht zeitkritische Anwendung ist, wurde die Verbindung auch nach dem Handover gehalten.

3.2.5 gnomemeeting (CVS-Version)

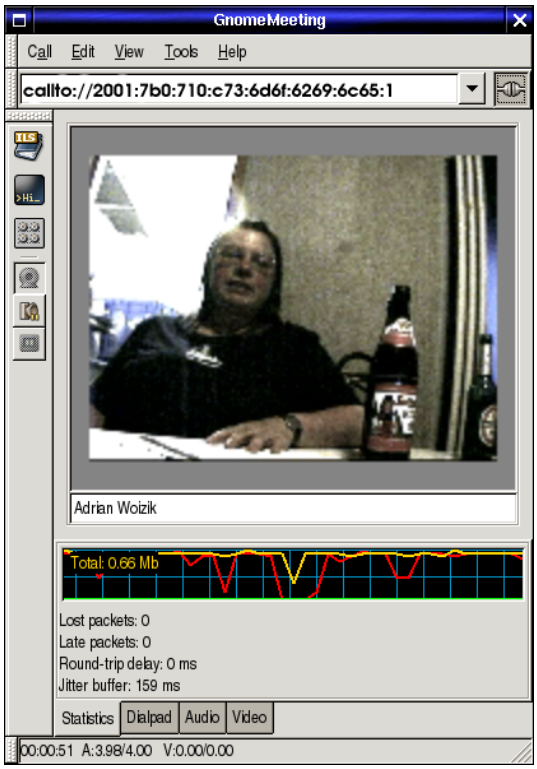

Abbildung 3.7: Gnomemeeting im Einsatz

Gnomemeeting[5] ist eine Videokonferenz Anwendung, die in der Standardversion nicht für IPv6 vorgesehen ist. Es gibt aber eine CVS-Version die man bequem als Debian-Paket installieren kann. Die von uns verwendete Version kann man von folgender Debian-Quelle installieren:

```
deb http://snapshots.seconix.com/debian/ sid main
```

Da diese Anwendung sehr bandbreitenintensiv ist, und einen permanenten Stream rausschickt, war das Interesse groß, ob die Übertragung von Bild und Ton einem Netzwechsel standhält.

[5] http://snapshots.seconix.com/

Es wurde eine Verbindung zweier gnomemeeting Clients hergestellt über den HA **morpheus** hergestellt. Um die mobility der Verbindung zu testen wurde auf einem Client das Interface gewechselt, um einen Netzwerkwechsel zu simulieren. Wie bei der SSH Verbindung wurde wieder erwartet, dass sie bestehen bleibt. Leider war das nicht der fall. Die Anwendung hat sich als sehr zeitkritisch gezeigt und hat die 1,5 Sekunden des Routenwechsels nicht immer überstanden. Es hat sich öfter der Fall rausgestellt in dem die Verbindung gekappt wurde. Machmal wurde noch ein eingehender Datenstrom verzeichnet und machmal nur ein ausgehender. Das hat uns gezeigt, dass Mobile IPv6 für Videokonferenzen noch nicht augereift ist.

3.3 Testvorgang und Testergebnisse

3.3.1 Testvorgang

Nach der beendeten Konzeptionierung wurde versucht jedes der einzelnen Versuche aufzubauen. Hierzu traf sich die Projektgruppe im Wohnheim Am Groshausberg.

- **WLAN Wechsel**
  Um den Wechsel in ein anderes WLAN, welches ein unterschiedliches IPv6 Subnetz besitzt, zu vollziehen wurde ein kleines shell script verwendet. Da Linux noch nicht in der Lage ist selbständig nach neuen Netzen zu suchen musste dieser Wechsel manuel erfolgen.
  Hier bemerkten wir den Nachteil der derzeitigen Implementation des Handovers. Das System hat bei dem Wechsel einen neuen Router-Advertisment erhalten und am WLAN Interface eine neue Care-Of-Address konfiguriert. Dabei hat es aber die alte Care-Of-Address aus dem ursprünglichem Subnetz weiterhin am Interface gebunden gehabt. Beim Versuch eine Bindung Update Nachricht an den HomeAgent zu schicken, glaubte daher das System es könnte den HA noch am WLAN Interface erreichen. Diese fehlgeschlagenen Binding Updates waren dann der Grund das wir das Mobile-IPv6 Protokol mithilfe eines Interface Wechsels testen wollen.

- **Interface Wechel**
  Beim Interface Wechsel haben wir ebenfalls mittels einem shell Script manuell umgeschalten. Es wurde eine Gnomemeeting Session zwischen zwei Notebooks aufbaut und die Interface Statistiken beobachtet. Dabei wurde auch geschaut ob die Video-Übertragung ruckelfrei beim Wechsel zum Ethernet Interface bleibt. Um den Handover so reibungslos wie möglich abzuliefern, wurde auch eine Delay Zeit beim Wechsel eingebaut. So wurde zuerst das Ethernet Interface aktiviert und min. 2 sec gewartet bis das Interface auch ganz sicher eine Care-Of-Addresse besaß. Erst nach diesem Delay wurde das WLAN Interface disabled.

  Um im Vorfeld auf keine Probleme zu stossen wurden die Interfaces, die für diese Tests nicht benutzt wurden deaktiviert. Dazu kann z.B. bei Notebooks ein Infrarot oder Bluetooth Interface zählen. Im produktiven Betrieb besteht die Möglichkeit die Interfaces mit einer Priorisierung festzulegen die es einem erlaubt Ethernet und WLAN vor Interfaces mit niedrigerer Bandbreite zu bevorzugen.

3.3.2 Testergebnisse

- **Mobile IPv6**
  Der Mobile IPv6 Stack hat sich sowohl am HomeAgenten als auch auf den Notebooks nach einer nicht bestimmten Anzahl von Wechseln/Binding Updates aufgehängt. Hierbei wurde der IPv6 Stack komplett gestört so daß dieser auch beim unloaden des Mobile IPv6 Kernel Moduls keine Packete empfangen oder senden konnte.
  Nach einem `/etc/init.d/network restart` und ein neuladen des Mobile IPv6 Kernel Moduls konnten die Tests weitergehen.

- **ssh**
  Da SSH die unkritischste Applikation war, gab es hier auch die geringstens Probleme. Sofern sich der Mobile IPv6 Stack nicht aufgehängt hatte, blieb die Connection standhaft zwischen den Wechseln bestehen.

  Bei einer Kommunikation mit einem Corresponding Node wurde auch die Destination Option Variante genutzt. Der Binding Cache des Remote Hosts wurde Korreckt upgedated.

- **Gnomemeeting**
  Die Tests mit Gnomemeeting zählten zu den Intressantesten. Zu einem weil eine Video Konferenz eine Applikation ist die man sich sehr gut bei Mobilen Geräten in der Zukunft vorstellen kann und zu anderem weil durch die erhöte Packet-Rate Probleme beim Wechsel schneller auffallen.

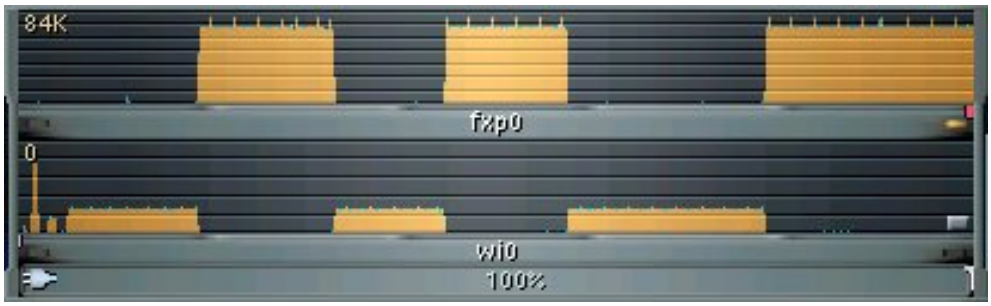

Abbildung 3.8: Traffic-Ansicht auf beiden Interfaces beim Wechsel

Wie in Abbildung 3.8 zu sehen ist wurden die Wechsel zwischen dem Ethernet Interface (fxp0) und dem WLAN Interface (wi0)[6] Problemlos absolviert. Die Video-Session wurde beim Wechsel jeweils am dafuer vorgesehenem Interface rausgeschickt. Die Binding Updates wurden Erfolgreich von der Gegenstelle (Eines der Notebooks) verarbeitet.

Der HomeAgent hat das Binding Update angenommen und Verbindungsanfragen über einen Tunnel an den Mobile Node gesendet. Der Mobile Node hat dann zuerst Versucht direkt zu Antworte und erst im Falle einer ICMP Nachricht – Die dem

[6] Aufgrund der Gewöhnung des Notebook Besitzers haben die Interfaces FreeBSD übliche Namensgebungen

Mobile Node zu verstehen gegeben hat das die Gegenstelle den Destination Option Header nicht verstehen kann – den Tunnel zur Kommunikation verwendet.

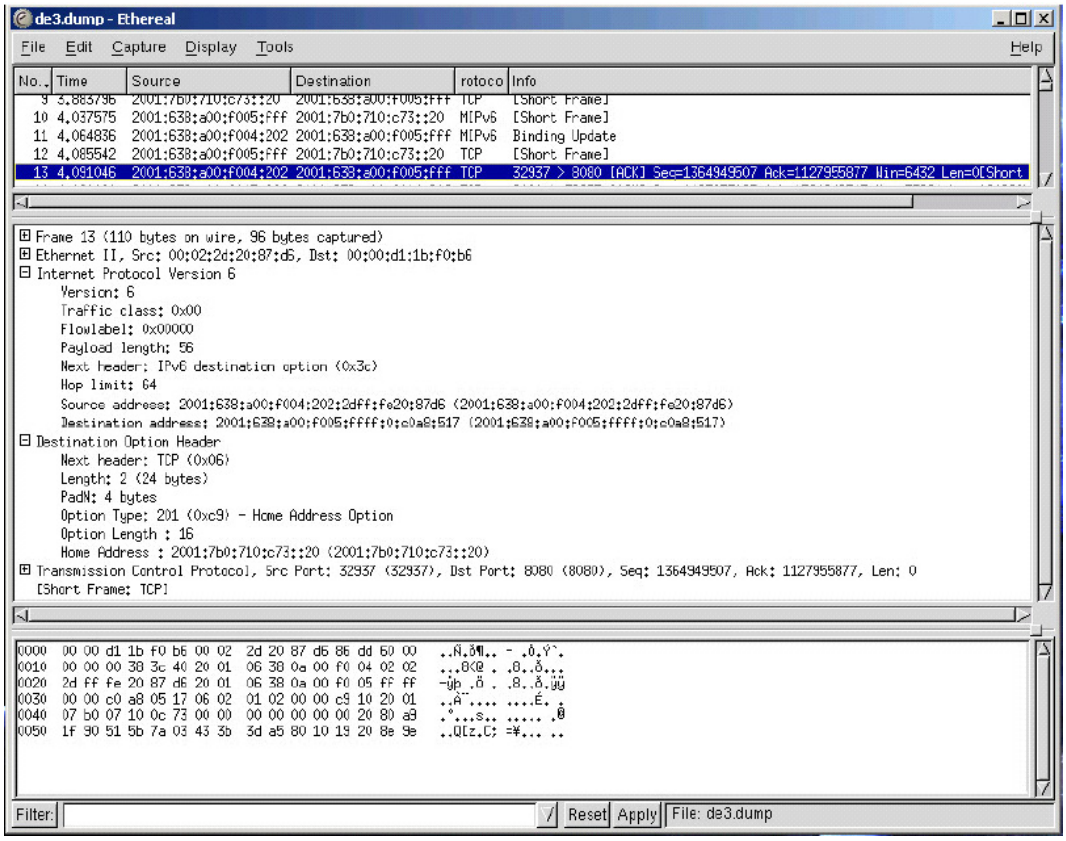

Abbildung 3.9: Erfolgreicher Mobile IPv6 Destination Option Header

Kapitel 4
Allgemeine Probleme

4.1 Sicherheitsüberlegungen

Die “binding update” Option wird den zu MN adressierten Paketen, die stattdessen zu CoA zugestellt werden, einverleibt. Diese Routeänderungsmöglichkeit von diesen Paketen könnte eine bedeutende Verwundbarkeit darstellen, wenn alle Pakete, die “binding update” Option enthalten, nicht authentifiziert sind. Die “binding acknowledgement” Option benötigt ebenfalls die Authentifizierung. Ein Angreifer könnte z.B. versuchen, MN auszutricksen und ihm vormachen, dass die kommende Nachricht von seinem HA stammt. Man braucht keine Authentifizierung für die “binding request” Option, wenn die Benutzung von dieser Option keine Änderungen des Status in Sender oder Empfänger hervorruft. Die “binding request” Option wirft einige Sicherheitsprobleme auf, die aber mit der Hilfe von IPSec Verschlüsselungsmechanismen oder durch Benutzung von Firewalls behoben werden können. IPSec wird derzeitig vom Linux Patch noch nicht unterstützt. Die in IPSec existierenden “replay protection” Mechanismen erlauben “replay protection window” die ankommenden Pakete nach “out of order” zu kontrollieren. Überwiegend müssen “binding updates” in derselben Reihenfolge ankommen, in der sie abgeschickt worden sind. Die “binding update” Option hat ein “sequence number” Feld, um die Reihenfolge kontrollieren zu können. IPSec unterstützt “strong replay protection”, aber keine Anordnung. “Sequence number” dagegen unterstützt die Anordnung, aber keine “replay protection”.

4.2 Debian Patch, Kernel Crash und Neuinstallation

Im Laufe unseres Pojekts begleiteten uns einige Probleme bezüglich des Kernels, so daß einige male auch eine Neukompilierung des Kernels notwendig war, sowie auch dessen Installation und natürlich eines folgenden Neustarts.

Die von uns Benutzte Kernel-Version war die Version 2.4.20 aus den Debian-Sourcen. Um Mobile-IPv6-Fähigkeiten zu erlangen wurde der Kernel vor der Kompilierung, mit dem von Debian gelieferten USAGI-Paket, gepatcht. Die Probleme die sich bei uns abzeichneten kamen bei einem Interface-Wechsel zum tragen. Sobald das Interface runtergefahren wurde konnte es bei unserem im Test verwedeten IBM-Notebook nicht wieder geladen

werden. Bei den beiden Compaq-Notebooks stürzte der Kernel mehrmals mit einer "Kernel Panic" ab und musste neu gebootet werden.

Der USAGI-Patch wurde bei Debian schon seit laegerer Zeit nicht aktualisiert, so lag es nahe einen aktuellen CVS-Snapshot des USAGI-Kernels von der USAGI-Homepage zu kompilieren. Das Angebotene STABLE Release des USAGI-Kernels war zu diesem Zeitpunkt schon mehrere Monate alt und somit für unsere Zwecke unbrauchbar. Das verwenden eines CVS-Snapshots eines Programms oder Kernels ist immer mit einem Restrisiko verbunden ob diese Version überhaupt lauffähig ist. Der Erhoffte Effeckt den wir mit dem USAGI-Kernel vebanden blieb jedoch aus, weitere Abstürze und Schwierigkeiten konnte auch über diesen Weg nicht vermieden werden. Unsere Privates IBM Notebook reagierte etwas stabiler auf den USAGI-Kernel als die Compaq Notebooks. Somit konnten wir zumindest mit einem MN mehr oder weniger stabile Test fahren. Der Ursache für die Compaq Abstürze und Probleme versucht der nächste Abschnitt auf den Grund zu gehen.

Die Probleme bezogen sich aber nicht nur auf die von Hewlett Packard gelieferten Notebooks, sondern auch auf die andere Eingesetzte Hardware. Nach unserer Erfahrung mit dem USAGI-Kit wagen wir zu behaupten das der ganze Mobile IPv6 Stack des Kernels noch sehr instabiel reagiert auf die kleinsten Veränderungen. Einfaches herunterfahren des WLAN-Interfaces auf verschiedenen Rechnern brachte eine Kernel Panic. Auch das herrausziehen der PCMCIA-WLAN Karte des IBM Notebooks versetzte das komplette System in einen instabilen Zustand mit dem sich nicht mehr arbeiten lies. Ein Binding Wechsel auf den Interfaces brachten den IPv6 Stack zum stehen, ohne Reboot liess sich das Problem nicht beheben weil man das IPv6 Kernel Modul nicht mehr entladen konnte.

4.3 Compaq USB-WLAN

Die von HP gelieferten Compaq N620c Notebooks verwenden die neueste Hardware, für die es unter Linux nur Treiber als beta-Version gibt. Die WLAN-Karte mit der Modell-Bezeichnung W200 basiert auf einem Orinoco-Chip. Dieser ist aber nicht wie die meisten Orinoco Karten am PCMCIA-Slot angeschlossen, sondern hier handelt es sich um eine USB-Version. Die von uns kompilierten und installierten Treiber[1] mögen im Alltag zwar sehr gut funktionieren, jedoch führten sie bei unseren Testverfahren, das auf einem Netzwerkwechsel zwischen dem WLAN-Netz und dem lokalen Netz per Ethernet-Karte beruht, zu permanenten Abstürzen des Kernels. Nach dem beenden des Projektes wurden neue Treiber Versionen für die USB-WLAN Karte released die wir aber zu dem späten Zeitpunkt nicht mehr in unsere Projekt einbinden und Testen konnten.

4.4 Betriebssystemabstimmung und Hardware

Um ein Reibungslosen Ablauf für das Mobile IPv6 Protokoll zu garantieren muss man die Vorhandene Hard- und Software perfekt auf einander abstimmen. In unserem Fall war dies nur begrenzt möglich das wir nicht die besten Vorraussetzung hatten. Die Probleme

[1] http://orinoco-usb.alioth.debian.org/

mit dem Compaq Notebooks wurden schon an anderer Stelle in dieser Dokumentation behandelt, deshalb wollen wir nicht noch einmal drauf eingehen. Zu beginn des Projektes stand uns die Möglichkeit zur Verfügung, einen Rechner im *foo Pool* der FH-Furtwangen zu benutzen und ihn mit einzubinden. Er sollte uns die Möglichkeit geben das Mobile IPv6 Protokoll in einem grösseren Rahmen zu testen, anstatt nur in dem Wohnheim. Am Anfang wurde uns ein System zur Verfügung gestellt das durch seine Hardware Fehler glänzte, es konnte kein stabiles System darauf installiert werden. Nach entsprechender Zeit konnten wir ein Austausch System bekommen. Leider stürtze dieser Rechner machmal ohne erkannbaren Grund öfters ab nachdem wir ein funktionierendes System drauf installiert hatten. Nach der vielen Arbeit die wir in die beiden Rechner gesteckt hatten entschied sich die Projekt Gruppe das Komplette Test Szenario nur im Wohnheim afzubauen und testen.

Zusätzlich zu den Fremdrechnern kamen auch noch Probleme mit den eigenen Rechnern zum tragen, auf unserem FreeBSD Correspondant Node **realworld** fiel mitten in den Testphasen die System Festplatte aus und der Rechner musste neu Aufgesetzt werden.

Kapitel 5
Aufwand / Kenntnisstand

5.1 Kenntnisstand

5.1.1 des Anwenders

Unter dem Anwender versteht man den Standard Benutzer, der sich nur mit dem benutzen eines Systems und dessen Anwendungen beschäftigt und keinerlei Änderungen an der Konfiguration vornimmt.

Der User muss ein durchschnittlichen Kenntnisstand des Betriebssystems haben und mit den Grundlagen von Mobile IP vertraut sein. Dies bezieht sich natuerlich nur auf ein sehr gut konfiguriertes System, das es nicht erfordert das der Benutzer in die Konfiguration eingreiffen muss. Sollte dies nicht der Fall sein, wird ein höherer Wissenstand vorrausgesetzt um sich in der IPv6 Welt mobile zu bewegen.

5.1.2 des Administrators

Momentatn sindnoch keine Standard Programme vorhanden, die es ermöglichen ein mobiles IPv6 System einzurichten. daher benötigen wir zusätzliche Kernel Patches, die nur von erfahrenen Usern eingespielt und konfiguriert werden können. Zusätzlich sollte sich der Administrator mit dem Betriebssystemen und Netzwerken auseinandersetzen können. Der Administrator sollte auch auf dem neusten Stand der Technik sein und sich regelmäßig über neue Erweiterungen informieren, um die Performance und die Stabilität des Systems zu steigern. Die Ganze Mobile IPv6 Welt ist derzeit noch im Entwicklungsstadium und bedarf daher noch einer sehr hoher Experimentierfreudigkeit. Die Microsoft[1] Welt ist noch weit von Mobile IPv6 entfernt, obwohl es einige Ansätzegab, die von Microsoft aber derzeit nicht mehr weiterentwickelt werden.

[1] http://www.microsoft.com

5.2 Aufwand

5.2.1 Installation / Konfiguration

Die Installation ist je nach Kenntnisstand unterschiedlich. Erfahrene User kommen deutlich besser damit zurecht. Eine Standard Debian (unstable) Installation setzen wir voraus und zählen sie somit nicht zum tatsächlichem Aufwand dazu. Die Installation eines neuen Kernels ist im Vergleich schon ein grössere Aufwand, besonders wenn man sich mit der Hardware des Systems nicht auskennt. Zusätzlich es es eine grössere Umstellung das Betriebssystem auf das IPv6 Protokoll einzustellen. Im grossen und ganzen kann man den Aufwand in diesem Moment nur als sehr hoch betrachten.

FIXME: Tabelle aus Word Doc einfuegen.

5.2.2 Administration eines bestehenden Netzes

Da wir noch kein stabiles Netz haben, ist der administrative Aufwand extrem hoch, jedoch sind wir uns sicher daß der Aufwand mit einer höheren Stabilität des Systems deutlich abehmen wird. Mit der Zeit wird sich der Aufwand von den Grundlagen (Stabilität des System) zu den Bereichen Security und Performance verschieben.

Kapitel 6
Fazit

6.1 Stand des Projektes zum Abschluss

Durch die Verschiedensten Hardware Probeme waren wir gezwungen auf grössere Test mit dem Protokoll zu verzichten. Grundlagen Test konnten verwirklicht werden aber nur mit Einschränkungen die in der ganzen Dokumentation beschrieben wurden. Zu Ende des Projektes konnten wir die meisten Probleme aus dem Weg räumen und mit mehr Zeit und den gewonnen Erfahrungen sind wir uns sicher das wir noch mehr und bessere Ergenisse erziehlt hätten.

!FIXME! keine ahnung mehr was ich labern kann

- 2 Router mit 2 getrennten Netzen. FreeBSD / Debian
- 1 Notebook HP Linux (Debian unstable) mit Windows 2000/XP
- 2 Notebooks Compaq evo 610c
- 1 Notebook IBM Thinkpad T23, linux debian unstable

6.2 Resume

Die Mobilität in einem IPv6 Netzwerk ist ein sehr Intressantes Thema und hat eine Menge Potenzial. Es stellt uns gute Möglichkeiten zu Verfügung wie man den Wechsel eines Mobile Node in Verschiedenen Netzwerken erfolgreich bewerkstelligen könnte. Sobald das Mobile IPv6 Protokoll mal aus den Kinderschuhen entwachsen ist sollte man ein zweites mal seinen Blick drauf werfen. Zu diesem Zeitpunkt kann man noch keine genauen Aussagen über das Protokoll machen so wie wr es testen konnten. In einer Idealeren Umgebung würde die Wertung vielleicht anderst ausfallen. Aber ein perfektes Versuchslabor stand uns einfach nicht zur Verfügung. Da ja auch die grosse Umstellung auf IPv6 erst so langsam beginnt besteh noch kein Druck auch gleich mit der Mobilen *Tür* ins Haus zu fallen. Es ist auch ein Grund wieso der Aufwand noch zu hoch ist, da noch keine flächendeckenden IPv6 Netze in Deutschland vorhanden sind. Unser Test bewiesen mit

unserem Test Feld das man noch keinen stabilen Wechsel des Netzwerkes mit der Vorhanden Soft- und Hardware vernünftig hinbekommt. Wir erwarten das es noch eine längere Zeit braucht bis das Mobile IPv6 Protokoll voll ausgereift ist und der Durchbruch zuerst auf dem Mobile Phone Sektor zuerst kommt. Mit dem entsprechendem Einsatz von genügend Resourcen könnte man die Entwicklung schneller vorrantreiben und besser Ergebnisse bei weiteren Test erziehlen.

Glossar

B

Bandbreite
Übertragungskapazität eines Links in Bits/s.

Binding
Bezeichnet die Verbindung der Heimatadresse eines mobilen Knotens mit seiner Care-of-Adresse.

Binding Cache
Enthält Informationen über aktuelle Care-of-Addressen. Der Mobile IPv6 Stack überprüft vor der Routing-Entscheidung ob die Ziel IPv6 Addresse eine Care-of-Addresse besitzt über die sie die HomeAddresse erreichen kann.

C

Care-of-Adresse (CoA)
Eine IP-Adresse die einem mobilen Knoten zugewiesen ist, wenn er einen foreign link besucht. Ein mobiler Knoten kann mehrere care-of-adressen besitzen. Diejenige, die bei dem Home Agent des Knotens registriert ist, nennt man primary Care-of-Adresse.

Correspondent Node (CN)
Ein Knoten mit dem ein mobiler Knoten kommuniziert. Dieser Knoten kann sowohl stationär als auch mobil sein. Der Knoten muss Destination Option Header auswerten können.

D

Destination Option Header
IPv6 Erweiterungsheader der bei Mobile IPv6 dazu benutzt wird Bindings an Packete dran zuhaengen.

E

EUI64
Stateless Autoconfiguration in Ethernet basierenden Netzen. Definiert in RFC2464.

F

Foreign Agent (FA)
Ist ein Router, der dem mobilen Knoten in einem fremden Netzwerk Routing Services bietet. Er entpackt und vermittelt getunnelte Pakete, die er vom Home Agent des mobilen Knotens erhält. Für ausgehende Pakete dient er dem mobilen Knoten als Default Router.

Frame
Datenpaket auf OSI-Layer 2.

H

Home Agent (HA)
Ein Router, der sich im Heimatnetzwerk des mobilen Knotens befindet und bei dem er seine aktuelle Care-of- Adresse registrieren lässt. Befindet sich der mobile Knoten nicht in seinem Heimatnetz, dann fängt der Home Agent seine Pakete ab, kapselt sie in eine andere Nachricht und tunnelt sie an die primäre Care-of-Adresse.

L

LAN
Local Area Network, lokal begrenztes Netz mit hohen Übertragungsraten

link-local
Ein Packet kann direkt zum Ziel geschickt werden da das Ziel sich im selben Subnetz sich befindet

M

Mobile Node (MN)
Ein Knoten, der seine Anbindung von einem Link zu einem anderen wechseln kann und weiterhin unter seiner Heimatadresse (home address) zu erreichen ist.

N

Node
Ein Knoten (Gerät) der IP implementiert (z.B. ein Host oder Router).

P

Paket
Datenpaket auf OSI-Layer 3.

R

Round-Trip Time (RTT)
Paketlaufzeit von Sender zum Empfänger und zurück. Massgeblicher Parameter für die Qualität interaktiver Services. Ausserdem Parameter für die Flusssteuerung von TCP.

T

Tunneln
Ein IP-Paket wird in das Nutzdatenfeld eines neuen IP-Paketes gesteckt und verschickt. Liest der Empfänger die Nutzdaten des empfangenen Paketes aus, so erhält er das ursprüngliche Paket.

W

WAN
Wide Area Network

WLAN
Wireless Local Area Network, basierend auf 802.11b

Hardware

.1 Mobile Node:

Compaq Notebooks:

Technische Daten	
Prozessortyp	Intel Pentium-M
Prozessortakt (MHz)	1400
Installiertes RAM (MByte)	256
LAN-Verbindung	Gigabit-Ethernet
WLAN-Karte	Wireless Lan W200 Orinoco USB

Betriebssystem:

- Debian GNU sid unstable
- CVS Snapshot von http://www.linux-ipv6.org/
- USAGI(UniverSAl playGround for Ipv6) Project Verfügbar unter: ftp://ftp.linux-ipv6.org/pub/usagi/snap/kit/
- Weitere Software: Gnomeeting von http://www.gnomemeeting.org Verfügbar als Debian Source: deb http://snapshots.seconix.com/debian/ sid main

IBM Notebook:

Technische Daten	
Prozessortyp	Intel Pentium-Mobile
Prozessortakt (MHz)	1133
Installiertes RAM (MByte)	512
LAN-Verbindung	10/100MBit-Fast Ethernet
WLAN-Karte	Orinoco Gold PCMCIA

Betriebssystem:

- Debian GNU sid unstable
- CVS Snapshot von http://www.linux-ipv6.org/ USAGI(UniverSAl playGround for Ipv6) Project Verfügbar unter: ftp://ftp.linux-ipv6.org/pub/usagi/snap/kit/
- Weitere Software: Gnomeeting von http://www.gnomemeeting.org Verfügbar als Debian Source: deb http://snapshots.seconix.com/debian/ sid main

.2 Home Agent:

Morpheus:

Technische Daten	
Prozessortyp	AMD-K6(tm) 3D processor
Prozessortakt (MHz)	300
Installiertes RAM (MByte)	256
LAN-Verbindung	10/100MBit-Fast Ethernet
WLAN-Karte	D-Link DWL 900 AP+ (Access Point)

Betriebssystem:

- Debian GNU sid unstable
- CVS Snapshot von http://www.linux-ipv6.org/ USAGI(UniverSAl playGround for Ipv6) ProjectVerfügbar unter: ftp://ftp.linux-ipv6.org/pub/usagi/snap/kit/
- Weitere Software: radvd (Router Advertisment Daemon) enthalten in Usagi Packet

.3 Corresponding Node:

Realworld:

Technische Daten	
Prozessortyp	Intel Pentium2
Prozessortakt (MHz)	400
Installiertes RAM (MByte)	512
LAN-Verbindung	4 Port D-Link Fast Ethernet Karte
WLAN-Karte	Orinoco AP 500

Betriebssystem:

- Free BSD 4.8 STABLE
- CVS Snapshot von http://www.kame.net/
- Weitere Software: radvd (Router Advertisment Daemon) enthalten in Kame Packet

Kernel Konfiguration

```
#
# Code maturity level options
#
CONFIG_EXPERIMENTAL=y

#
# Loadable module support
#
CONFIG_MODULES=y
CONFIG_MODVERSIONS=y
CONFIG_KMOD=y

#
# Networking options
#
CONFIG_PACKET=y
# CONFIG_PACKET_MMAP is not set
# CONFIG_NETLINK_DEV is not set
CONFIG_NETFILTER=y
CONFIG_NETFILTER_DEBUG=y
# CONFIG_FILTER is not set
# CONFIG_NET_NEIGH_DEBUG is not set
# CONFIG_NET_RESTRICTED_REUSE is not set
CONFIG_UNIX=y
CONFIG_INET=y
# CONFIG_IPSEC is not set
# CONFIG_IP_MULTICAST is not set
# CONFIG_IP_ADVANCED_ROUTER is not set
# CONFIG_IP_PNP is not set
CONFIG_NET_IPIP=y
# CONFIG_NET_IPIP_IPV6 is not set
# CONFIG_NET_IPGRE is not set
# CONFIG_ARPD is not set
# CONFIG_INET_ECN is not set
# CONFIG_SYN_COOKIES is not set
CONFIG_IPV4_IPSEC_TUNNEL=y

#
#   IP: Netfilter Configuration
#
CONFIG_IP_NF_CONNTRACK=y
CONFIG_IP_NF_FTP=y
CONFIG_IP_NF_IRC=y
CONFIG_IP_NF_QUEUE=y
CONFIG_IP_NF_IPTABLES=y
CONFIG_IP_NF_MATCH_LIMIT=y
CONFIG_IP_NF_MATCH_MAC=y
CONFIG_IP_NF_MATCH_PKTTYPE=y
CONFIG_IP_NF_MATCH_MARK=y
```

```
CONFIG_IP_NF_MATCH_MULTIPORT=y
CONFIG_IP_NF_MATCH_TOS=y
CONFIG_IP_NF_MATCH_ECN=y
CONFIG_IP_NF_MATCH_DSCP=y
CONFIG_IP_NF_MATCH_AH_ESP=y
CONFIG_IP_NF_MATCH_LENGTH=y
CONFIG_IP_NF_MATCH_TTL=y
CONFIG_IP_NF_MATCH_TCPMSS=y
CONFIG_IP_NF_MATCH_HELPER=y
CONFIG_IP_NF_MATCH_STATE=y
CONFIG_IP_NF_MATCH_CONNTRACK=y
CONFIG_IP_NF_MATCH_UNCLEAN=y
CONFIG_IP_NF_MATCH_OWNER=y
CONFIG_IP_NF_FILTER=y
CONFIG_IP_NF_TARGET_REJECT=y
CONFIG_IP_NF_TARGET_MIRROR=y
CONFIG_IP_NF_NAT=y
CONFIG_IP_NF_NAT_NEEDED=y
CONFIG_IP_NF_TARGET_MASQUERADE=y
CONFIG_IP_NF_TARGET_REDIRECT=y
CONFIG_IP_NF_NAT_LOCAL=y
CONFIG_IP_NF_NAT_SNMP_BASIC=y
CONFIG_IP_NF_NAT_IRC=y
CONFIG_IP_NF_NAT_FTP=y
CONFIG_IP_NF_MANGLE=y
CONFIG_IP_NF_TARGET_TOS=y
CONFIG_IP_NF_TARGET_ECN=y
CONFIG_IP_NF_TARGET_DSCP=y
CONFIG_IP_NF_TARGET_MARK=y
CONFIG_IP_NF_TARGET_LOG=y
CONFIG_IP_NF_TARGET_ULOG=y
CONFIG_IP_NF_TARGET_TCPMSS=y
CONFIG_IP_NF_ARPTABLES=y
# CONFIG_IP_NF_ARPFILTER is not set
CONFIG_IPV6=m
# CONFIG_IPV6_DEBUG is not set
CONFIG_IPV6_IM=y
CONFIG_IPV6_ZONE=y
CONFIG_IPV6_ZONE_SITELOCAL=y
CONFIG_IPV6_DROP_FAKE_V4MAPPED=y
# CONFIG_IPV6_RESTRICTED_DOUBLE_BIND is not set
# CONFIG_IPV6_6TO4_NEXTHOP is not set
CONFIG_IPV6_PRIVACY=y
# CONFIG_IPV6_ANYCAST is not set
# CONFIG_IPV6_ISATAP is not set
# CONFIG_IPV6_PREFIXLIST is not set
CONFIG_IPV6_SUBTREES=y
CONFIG_IPV6_ROUTER_PREF=y
CONFIG_IPV6_NEW_ROUNDROBIN=y
# CONFIG_IPV6_ROUTE_INFO is not set
# CONFIG_IPV6_MLD6_ALL_DONE is not set
# CONFIG_IPV6_NODEINFO is not set

#
#   IPv6: Netfilter Configuration
#
# CONFIG_IP6_NF_QUEUE is not set
# CONFIG_IP6_NF_IPTABLES is not set
CONFIG_IPV6_IPSEC_TUNNEL=y
CONFIG_IPV6_IPV6_TUNNEL=m
CONFIG_IPV6_MOBILITY=m
CONFIG_IPV6_MOBILITY_MN=y
CONFIG_IPV6_MOBILITY_DEBUG=y
# CONFIG_KHTTPD is not set
# CONFIG_ATM is not set
# CONFIG_VLAN_8021Q is not set
# CONFIG_IPX is not set
```

```
117 # CONFIG_ATALK is not set
118
119 #
120 # Appletalk devices
121 #
122 # CONFIG_DEV_APPLETALK is not set
123 # CONFIG_DECNET is not set
124 # CONFIG_BRIDGE is not set
125 # CONFIG_X25 is not set
126 # CONFIG_LAPB is not set
127 # CONFIG_LLC is not set
128 # CONFIG_NET_DIVERT is not set
129 # CONFIG_ECONET is not set
130 # CONFIG_WAN_ROUTER is not set
131 # CONFIG_NET_FASTROUTE is not set
132 # CONFIG_NET_HW_FLOWCONTROL is not set
133
134 #
135 # QoS and/or fair queueing
136 #
137 # CONFIG_NET_SCHED is not set
138
139 #
140 # Network testing
141 #
142 # CONFIG_NET_PKTGEN is not set
143
```

Literaturverzeichnis

[AC01] Alan Clarkson, Paul Kummer, Robin Tasker. *A Project to Investigate the Effectiveness of QoS Techniques*. http://icfamon.dl.ac.uk/I2QoS/final-report.pdf, 2001.

[Bon02] Boney, James. *Cisco IOS in a Nutshell*. O'Reilly & Associates Inc., 1. Auflage, 2002. ISBN 1-56592-942-x.

[Cis00a] Cisco. *Catalyst 6000 and 6500 Series Software Configuration Guide*. Technischer Bericht, Cisco, 2000.

[Cis00b] Cisco. *Troubleshooting Catalyst Switches (Part 2)*, 2000.

[JD03] Johnson D, Perkins C. *Mobility Support in IPv6 draft-ietf-mobileip-ipv6-24.txt*. http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-24.txt, 2003.

[Moo02] Moore, David, Keys, Ken, Koga, Ryan, Lagache, Edouard, und k. claffy. *The CoralReef software suite as a tool for system and network andministrators*. www.caida.org/outreach/papers/2001/CoralApps/CoralApps.pdf, 2002.

[Sav99] Savage, Stefan. *Sting: a TCP-based Network Measurement Tool*. http://www.cs.washington.edu/homes/savage/papers/Usits99.pdf, 1999.

[Sie00] Siemens, Edgar. *Realisierung und Bewertung eines QoS-Dienstes im Campus-Netz*. Diplomarbeit, Universität Hannover, Lehrgebiet Rechnernetze und Verteilte Systeme, 2000.

[Ste94] Stevens, Richard. *TCP/IP Illustrated, Vol. 1*. Addison-Wesley, 17. Auflage, 1994. ISBN 0-201-63346-9.

[Sto99] Stoy, Robert und Jähnert, Jürgen. *Test of CISCO's IP QoS Implementation considering Differntiated Services*. http://www.cnaf.infn.it/~ferrari/tfng/doc/ds/dstest-unist-v0.9.doc, 1999.

[Tan98] Tanenbaum, Andrew S. *Computernetzwerke*. Prentice Hall, 1998. ISBN 3-8272-9568-8.

[Wan01] Wang, Zheng. *Internet QoS, Archtitectures and Mechanisms for Quality of Service*. Morgan Kaufmann Publisher, 2001. ISBN 1-55860-608-4.

[Web00] Webb, Karen. *Multi Layer Switched Netzwerke*. Markt+Technik Verlag, München, 2000. ISBN 3-8272-5854-5.