Mobile-IP
Einrichtung einer Testumgebung unter Linux

Fortgeschrittenenpraktikum am Lehrstuhl Prof. Dr. Heinz-Gerd Hegering

Bearbeiter: Thomas Lankes
Betreuer:Stephen Heilbronner

20. Juni 1996

Inhaltsverzeichnis

1 Einleitung/Motivation 3

2 Beschreibung von Mobile IP 4

2.1 Begriffserklärung 4

2.2 Funktionsweise von Mobile-IP 4

2.3 Advertisement und Solicitation 6

2.4 Authentifizierung 7

2.5 Registrierung 7

2.5.1 Registrierung via FA 7

2.5.2 Registrierung direkt mit dem HA 7

3 Implementierungen von Mobile-IP 9

3.1 Übersicht 9

3.2 Implementierung der University of Singapore 9

3.2.1 Funktionalität 9

3.2.2 Momentane Situation 10

4 Die Testumgebung 11

4.1 Installation von Linux 12

4.1.1 Ablauf der Installation 12

4.1.2 Besonderheiten bei der verwendeten Hardware 13

4.1.3 Kompilieren eines Kernels 15

4.2 Mobile-IP Erweiterungen 17

4.2.1 Installation der Kernel-Erweiterungen 17

4.2.2 Installation der Dämonprogramme 17
4.2.3 Konfigurationsdateien 18
4.2.4 Mobile-IP im Betrieb 22

5 Zusammenfassung/Ausblick 24

1 Einleitung/Motivation

Das zur Zeit verwendete Internet-Protokoll IPv4 [RFC1180] geht implizit davon aus, daß sich der Anschlußpunkt eines Computers nicht ändert. Falls ein Rechner in einem anderen Subnetz verwendet werden soll, so muß man entweder dem teilnehmenden Rechner eine andere Adresse zuweisen oder eine rechnerspezifische Route muß dem gesamten Internet Routing bekannt gemacht werden.

Beides sind vom Aufwand her inakzeptable Verfahren:

Ändert sich nämlich die Adresse eines Rechners im Netz, so muß dies im allgemeinen einer ganzen Reihe von anderen Geräten im Netz mitgeteilt werden. Rechner, Drucker und andere IO-Geräte die dem teilnehmenden Rechner das Recht einräumen deren Ressourcen zu nutzen, müssen über die neue Adresse informiert werden. Insbesondere, wenn ein portabler Rechner (Notebook) in verschiedenen Subnetzen verwendet werden soll, ist dieses Verfahren ebensowenig praktikabel, wie die zweite Möglichkeit, für das Notebook jeweils eine spzifische Route festzulegen.

Es ist also notwendig ein weiteres Verfahren zu entwickeln, das es erlaubt, einen Rechner in verschiedenen Subnetzen zu verwenden, ohne daß sich seine IP-Adresse ändert, bzw. ohne daß aufwendiges Umkonfigurieren notwendig wird.

Ein möglicher Ansatz ist Mobile-IP [PERK96]. Ziel von Mobile-IP ist es, daß ein Benutzer immer in seier gewohnten Umgebung arbeit (d.h. seine IP-Adresse ändert sich nicht) und dies in verschiedenen Subnetzen möglich ist.

Die Internet Engineering Task Force (IETF) hat in einem Internet-Draft [PERK96] das Mobile-IP-Protokoll spezifiziert, welches ich im folgenden näher beschreiben werde.

2 Beschreibung von Mobile IP

2.1 Begriffserklärung

In [PERK96] werden einige Begriffe verwendet, welche im Vorfeld beschrieben werden sollten:

Mobile Node (MN): Der Rechner, der vom Benutzer mobil verwendet werden soll. Die IP Adresse des MN darf sich nicht ändern, obwohl der MN in verschiedenen Subnetzen angeschlossen werden kann.

Home Agent (HA): Ein Rechner in dem Subnetz, in dem der Benutzer üblicherweise arbeitet. Der HA leitet den gesammten Netzverkehr für den MN an das Subnetz weiter, in welchem sich der MN zur Zeit befindet.

Foreign Agent (FA): Ein Rechner in dem Subnetz, in dem der MN zur Zeit angeschlossen ist. Datagramme, die der FA vom HA empfängt, werden vom FA an den MN weitergeleitet. Der FA fungiert für MN als Router.

Care-of address: Die Care-of address definiert das Ende eines Tunnels, in dem die den MN betreffenden Datagramme vom HA zum MN weitergeleitet werden. Abhängig von der Konfiguration des Netzes bezieht sich die Care-of address entweder auf den FA oder sie wird dem MN dynamisch zugewiesen (man spricht dann auch von einer Co-located Care-of address).

2.2 Funktionsweise von Mobile-IP

Wird Mobile-IP verwendet, so laufen bei den beteiligten Rechnern die folgenden Aktionen ab:

- Agenten verschicken Advertisements (siehe auch 2.3) (optional können Agenten vom MN, durch Solicitations, aufgefordert werden, Advertisements zu senden.

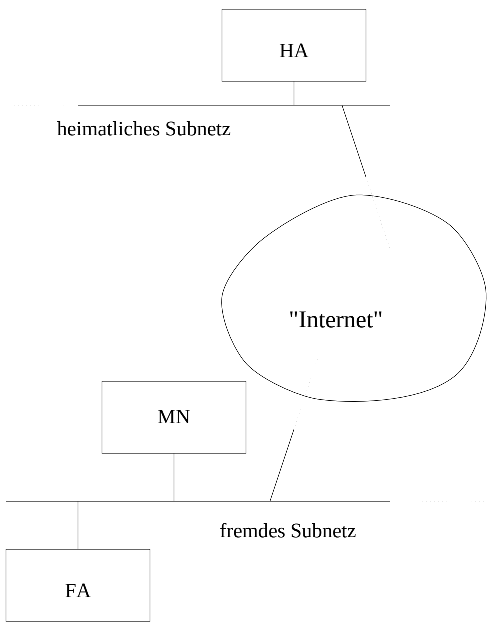

Abbildung 1: Der Mobile Node befindet sich im fremden Subnetz

- Ein MN empfängt Advertisements, und erkennt, ob er sich im heimatlichen Subnetz oder in einem fremden Subnetz befindet.
- Befindet sich der MN im heimatlichen Subnetz, so arbeitet der MN ohne die Agenten zu beachten weiter.

  Falls der MN wieder in sein heimatliches Subnetz zurückkommt, nachdem er in einem fremden Subnetz registriert war, so findet ein kurzer Nachrichtenaustausch zwischen MN und HA statt, der beide Rechner über die neue Situation unterrichtet.
- Befindet sich der MN in einem fremden Subnetz, so läuft eine Registrierung zwischen Agenten und MN ab.
- War die Registrierung erfolgreich, so werden Datagramme an den MN vom HA an die care-of address weitergeleitet. In umgekehrter Richtung kann der MN Datagramme ganz normal routen, diese müssen nicht unbedingt über den HA routen.

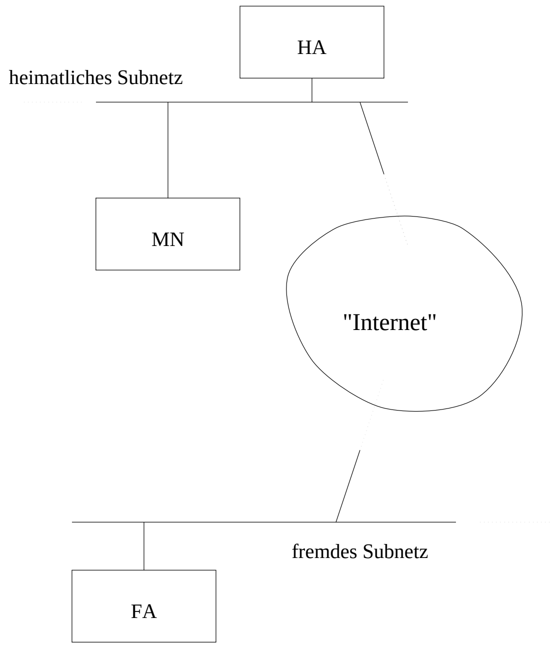

Abbildung 2: Der Mobile Node befindet sich im heimatlichen Subnetz

2.3 Advertisement und Solicitation

Advertisements werden von den Agenten verschickt. Es werden ICMP Router Advertisements [RFC1256] verwendet, welche um einige Informationen für den MN erweitert werden.

Empfängt der MN ein Advertisement, so erkennt er daran, ob er sich in seinem heimatlichen Subnetz oder in einem fremden Subnetz befindet.

Ein in einem Subnetz neu angekommener MN kann Solicitations senden. Diese dienen dazu, den MN über mögliche Agenten zu informieren.

Analog zu den Advertisements werden auch für die Solicitations ICMP Router Solicitation messages [RFC1256] verwendet.

2.4 Authentifizierung

In [PERK96] wird als Authentifizierungs-Mechanismus MD5 vorgeschieben (siehe auch [RFC1321]). Dabei handelt es sich um einen Algorithmus, der als Ausgabe einen „Fingerabdruck“ des Absenders produziert.

2.5 Registrierung

Soll der MN in einem fremden Subnetz verwendet werden, so findet zuerst eine Registrierung statt. Zweck ist es zum einen, die beteiligten Agenten über den MN zu informieren, und zum anderen das heimatliche Subnetz vor Angriffen zu schützen.

In [PERK96] werden zwei verschiedene Verfahren beschrieben, welche abhängig von der Konfiguration der Netze angewendet werden sollen.

2.5.1 Registrierung via FA

Handelt es sich bei der care-of address für den MN um die Adresse des FA, so wird in [PERK96] der folgende Austausch von Meldungen vorgeschrieben:

- Der MN sendet an den FA eine Meldung mit der Bitte den Ablauf der Registrierung zu beginnen.
- Der FA verarbeitet diese Anfrage, und leitet sie an den HA weiter.
- Je nachdem ob der HA den MN akzeptiert, beantwortet der HA die Anfrage und schickt an den FA eine Meldung.
- Nachdem der FA diese Meldung verarbeitet hat, informiert der FA den MN darüber ob der HA den MN akzeptiert hat.

2.5.2 Registrierung direkt mit dem HA

Ist die care-of address eine co-located care-of address im fremden Subnetz oder der MN soll wieder im Heimatlichen Subnetz verwendet werden, so schreibt [PERK96] den folgenden Vorgang vor:

- Der MN sendet eine Anfrage an den HA.
- Der HA beantwortet die Anfrage des MN.

3 Implementierungen von Mobile-IP

An der National University of Singapore wurde von Y.Z. Li, Y.C. Tay und K.C. Chuan eine Version von Mobile IP für Linux implementiert. Aufgabe des Fortgeschrittenenpraktikums war es, eine Testumgebung für diese Implementierung aufzubauen, die o.g. Implementierung von Mobile-IP zu installieren und zu testen ob und inwieweit es praktikabel ist mit Mobile IP zu arbeiten.

3.1 Übersicht

Neben der bereits genannten Implementierung ist eine weitere Version von Mobile-IP für Linux zu erwähnen:

An der State University of New York befassen sich Abhijit Dixit, Vipul Gupta und Ben Lancki mit Mobile-IP für Linux. Momentan existiert eine Version 1.0. In der Dokumentation zur Version 0.95 [NYSTATE] wurde jedoch die Existenz eines Tunnels zwischen FA und HA vorausgesetzt, und der Benutzer muß die Wegewahltabelle der teilnehmenden Rechner selbst verändern.

3.2 Implementierung der University of Singapore

Bisher wurden zwei Versionen von der University of Singapore bereitgestellt.

Die erste Version (Dezember 95) war laut [SING], dem damaligen Draft (Version 13) gemäß implementiert. Da es sich aber als problematisch erwies, mit dieser Version zu arbeiten, diese überhaupt in Betrieb zu nehmen, wurde beschloßen, weitere Arbeiten mit der Version 1.0 durchzuführen.

3.2.1 Funktionalität

Laut [SING] bietet die jetzt aktuelle Version 1.0 die folgende Funktionalität:

- overlaping wireless cells
- mobile routers

- multiple wireless interfaces
- dynamic routing

Auch ist die jetzt aktuelle Version gemäß dem [PERK96] implementiert.

Mobile IP für Linux unterliegt der „GNU General Public Licence“, dies bedeutet, daß es jedermann erlaubt ist Mobile IP für Linux zu verwenden.

3.2.2 Momentane Situation

Nachdem eine Testumgebung für Mobile IP Institut für Informatik aufgebaut worden ist, wurden einige Veränderungen am Quellcode durchgführt.

Diese werden in weiteren Fortgeschrittenenpraktika dokumentiert. Auch wird im folgenden getestet, inwieweit eine Zusammenarbeit mit einer Implementierung von Manuel Rodriguez (Hewlett Packard Laboratories, Bristol in UK) möglich ist.

An diesem Test beteiligen sich auch einige Mitarbeiter an der Universität ENSTB - Departement Reseaux et Systemes Multimedia in Rennes.

Am Institut für Informatik der LMU München befaßt sich zur Zeit eine Diplomarbeit mit dem Management von Mobilen Endsystemen.

4 Die Testumgebung

Es war Aufgabe des Fortgeschrittenenpraktikums, auf drei Rechnern Linux zu installieren, die Kernelerweiterungen für Mobile IP vorzunehmen und einige Tests mit Mobile IP durchzuführen.

Da es für weiterführende Arbeiten mit Mobile IP eventuell notwendig ist, weitere Rechner mit Linux zu betreiben, sollte im Vorfeld auf die Installation von Linux im allgemeinen eingegangen werden.

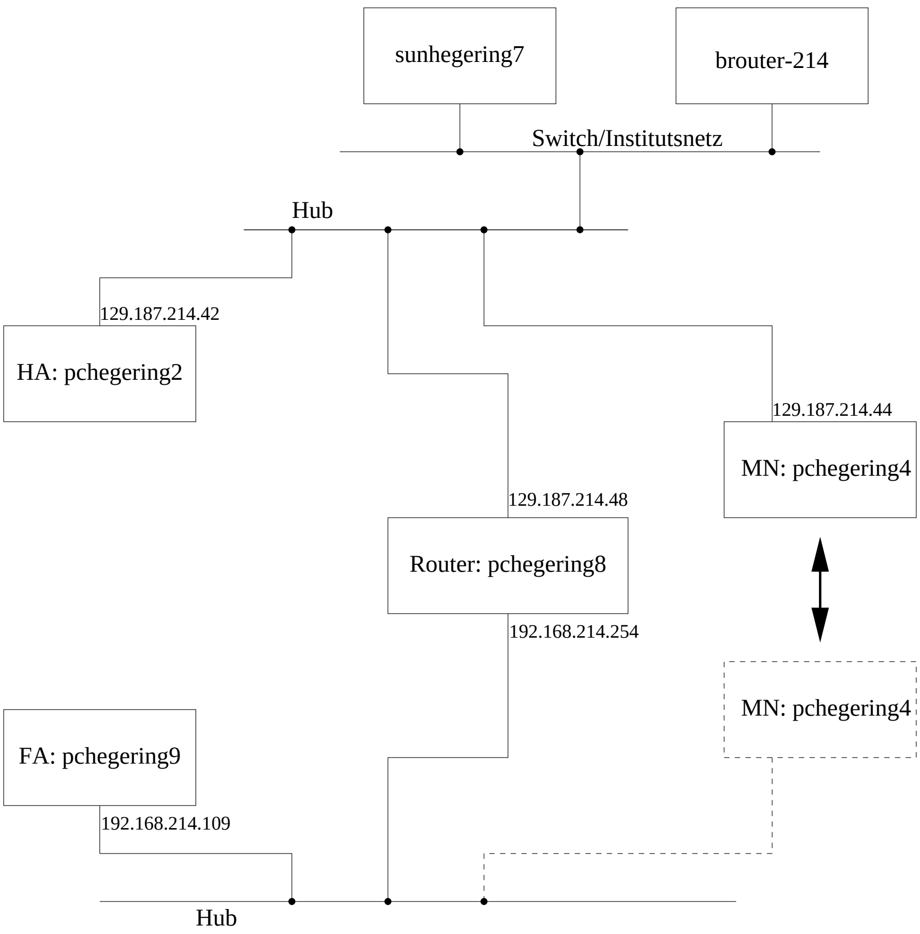

Abbildung 3: Testbed

4.1 Installation von Linux

Linux ist mittlerweile sehr gut dokumentiert (siehe z.B. [WEKA95] [KOFL95]), und dank einer einfach zu bedienenden Oberfläche gestaltet sich auch die Installation unproblematisch.

4.1.1 Ablauf der Installation

Aus Performancegründen ist es beim Betrieb von Linux, empfehlenswert, Linux auf eine eigene Partition zu installieren. Hierzu ist es eventuell notwendig, bereits vorhandene Partitionen (z.B. MS-Dos Partitionen) zu verkleinern. Das MS-Dos Programm presizer.exe kann MS-Dos Partitionen verkleinern, ohne daß eine vorhandene Partition gelöscht werden muß.

Als Distribution ist Slackware zu empfehlen, eine aktuelle Version befindet sich in [LEO].

Zuerst sollte man eine Boot- sowie eine Rootdiskette erstellen. Hierfür bietet sich das Programm rawrite.exe an (siehe auch [CHAP95])

Die Auswahl [1] der entsprechenden Image-Dateien erfolgt entsprechend der vorhandenen Hardware-Umgebung. Beim Booten mit der Bootdiskette wird unter anderem eine Ramdisk erzeugt, so daß dann ein minimales Linux System bereitsteht.

Sobald der Bootvorgang abgeschlossen ist, wird man aufgefordert, die Bootdiskette zu entfernen und die Rootdiskette in das Laufwerk einzulegen. Diese enthält die Installationsroutine.

Mit dem Program *fdisk* ist im Vorfeld eine genügend große Partition [2] für Linux zu erstellen. Falls der Hauptspeicher des Systems nicht ausreichend erscheint, so hat man an dieser Stelle auch die Möglichkeit eine *Swap-Partition*, zu erstellen.

Damit die mit *fdisk* vorgenommenen Änderungen wirksam werden, ist es notwendig, das System nochmal neu zu starten.

[1] ftp://ftp.leo.org/pub/comp/os/linux/slackware/bootdsks.144/WHICH.ONE

[2] soll auf einem Linux System X Windows betrieben werden, so kann man von einem Platzbedarf von über 200 MB ausgehen

Die Installationsroutine für Linux wird mit dem Befehl *setup* gestartet. Bevor mit der eigentlichen Installation begonnen werden kann, sind einige Einstellungen vorzunehmen, z.B. Auswahl des Quellmediums, der Zielpartition, welche Tastatur[3] verwendet werden soll und vor allem welche Teile der Distribution installiert werden sollen.

Innerhalb des Instituts für Informatik ist es möglich, Linux „über NFS“ zu installieren. Dabei sind die folgenden Angaben zu machen.

- Quellmedium: NFS
- Die eigene IP Adresse
- Die IP Adresse von [LEO]
- Der absolute Pfad, in dem sich die Slackware befindet:

  /archive/leo/.mntpts/tum.info-pub3/pub/comp/os/linux/slackware/slakware
- Gateway: Per Definition, x.y.z.254 (wobei x.y.z die ersten drei Byte der eigenen IP Adresse sind.

Sind die Einstellungen vorgenommen, so beginnt die eigentliche Installation.

Bevor jeweils ein Teil der Distribution auf die für Linux vorgesehene Partition kopiert wird, kann man nochmal im Detail angeben, welche Tools man verwenden will. Es ist jeweils beschrieben, was als nächstes kopiert wird und inwieweit es für den Betrieb notwendig ist.

Nachdem alle gewünschten Teile der Distribution auf das System kopiert wurden, sind nochmals einige Einstellungen vorzunehmen:

Es erscheinen nacheinander verschiedene Menüs, in denen man Linux z.B. über ein Modem, eine Maus bzw. die Zeitzone in der man sich befindet, informieren sollte.

4.1.2 Besonderheiten bei der verwendeten Hardware

Die meisten Probleme entstehen dadurch, daß einzelne Komponenten der Hardware nicht unterstützt werden. Da es durchaus möglich ist, daß auf

[3] als Tastaturtreiber für deutsche Tastaturen ist de-latin1 zu empfehlen

ähnliche bzw. gleiche Geräte für weitere Tests mit Mobile-IP ebenfalls Linux zu installieren ist, erscheint es mir angebracht, kurz auf die Besonderheiten der verwendeten Rechner und der bereitgestellte Hardware einzugehen.

Linux auf eine HP Vectra XU (pchegering2) installieren:

Die von der Slackware bereitgestellten Bootdisketten booten Linux 1.2. Jedoch ist eine HP Vectra XU mit einem AM53/79C974 SCSI Controller ausgestattet.

Da ein AM53/79C974 SCSI Controller erst ab Linux 1.3.x [4]. unterstützt wird, ist es notwendig, eine passende Bootdiskette, mit einem anderen Rechner zu erstellen.

Vorgehen:

- Den gewünschten Kernel Quellcode auf einen anderen Linuxrechner kopieren.
- Erweiterungen für einen AM53/79C974 SCSI Controller einbinden.
- wie in 4.1.3 und der Datei [CHAP95]

  beschrieben, eine Bootdiskette erstellen.

Von da an kann man wie bei einer normalen Installation vorgehen.

Linux auf HP Vectra QS installieren (pchegering8/pchegering9

Die beiden Rechner HP Vectra QS wurden je einmal als Router und einmal als FA konfiguriert.

Leider waren die bereitgestellten Rechner nur mit einer sehr gering bemessenen Festplatte ausgestattet. Da aber bereits der entpackte Quellcode für Linux in der Version 1.3.42 über 13 MB umfaßt, und es für den Betrieb von Linux unumgänglich ist, mindestens Teile des Quellcodes dem System zur Verfügung zu stellen, wurde Folgendes realisiert:

Das Verzeichnis /usr/src/linux des pchegering2 wurde beiden Rechnern über NFS zur Verfügung gestellt. Hierfür waren folgende Änderungen an Konfigurationsdateien der beteiligten Rechnern nötig:

[4] ab Linux 1.3.59 sind die Erweiterungen für einenAM53/79C974 SCSI Controller bereits enthalten

pchegering2: /etc/export

Einfügen der Zeilen:

```
/usr/src/linux  129.187.214.48(rw)
/usr/src/linux  192.168.214.109(rw)
```

pchegering8/pchering9: /etc/fstab

Einfügen der Zeile:

```
129.187.214.42:/usr/src/linux /usr/src/linux  nfs     defaults
```

Linux auf ein Highscreen Notebook installieren

Da das verwendete Highscreen Notebook nicht mit einer Ethernet-Karte ausgestattet war, mußte ein sog. D-Link Adapter verwendet werden.

Es handelt sich dabei um ein Gerät das an der parallelen Schnittstelle angeschloßen werden kann und dann eine Anschlußmöglichkeit für Twisted Pair bietet.

Bei der Verwendung eines D-Link Adapters ist jedoch folgendes beachtenswert:

Will man einen Treiber für einen D-Link Adapter in einen Kernel integrieren, so ist es entgegen der Dokumentation [5] der Treibersoftware auch notwendig, während *make config* die Frage, ob man Pocket Adapter verwenden will, mit ja zu beantworten. Nur dann erscheint die Auswahlmöglichkeit für die verschiedenen D-Link Varianten.

Auch legt der D-Link Adapter mitunter im Betrieb ein merkwürdiges Verhalten an den Tag. Manchmal erscheint beim booten von Linux dir Meldung *bogus interrupt 7*. Diese beeinflußt den Betrieb jedoch nicht. Leider tritt dieser Fehler auch nicht deterministisch auf, so daß es nicht möglich war, dafür eine Ursache zu finden, bzw. ihn zu beheben.

4.1.3 Kompilieren eines Kernels

Wie bei jedem Programm, hat man auch bei Linux die Wahl zwischen verschiedenen Kernel-Versionen. Bei Linux schreitet die Entwicklung sehr schnell

[5] Datei /usr/src/linux/drivers/net/README.DLINK

voran. Man könnte fast monatlich eine neue Version installieren.

Eine Version von Linux wird durch drei Zahlen spezifiziert. Die erste Zahl bedeutet die Release im allgemeinen. Die zweite Zahl gibt Auskunft darüber, ob es sich um eine anerkannt stabile Version handelt, oder ob es ein so genannter *„Hacker-Kernel"* ist. Die dritte Zahl letztlich wird fortlaufend (bei den Hacker-Kernels fast täglich) vergeben.

Da Mobile-IP den Verkehr zwischen den Agenten über das IP in IP Protokoll abwickelt (s.g. tunneling), und dies erst in Versionen 1.3.x bereitsteht, ist es unumgänglich einen der Hacker Kernels zu verwenden. Bei der Arbeit an diesem Fortgeschrittenenpraktikum, hat sich Linux 1.3.42 und auch Linux 1.3.59 als durchaus akzeptabel und stabil erwiesen.

Auch die aktuellen Versionen von Linux befinden sich in [LEO]. Das genaue Vorgehen, bei der Installation einer neuen Kernel Version wird in [WARD95] beschrieben.

1. Löschen des kompletten Verzeichnisses */usr/src/linux*.
2. Entpacken des Kernel-Quellcodes im Verzeichnis */usr/src*.
3. Wie in der Datei */usr/src/linux/README* beschrieben, müßen einige Links auf Verzeichnisse im neuen Quellcode aktualisiert werden.
4. Der wichtigste Teil des Vorgangs ist *make config*: Ausgeführt im Verzeichnis */usr/src/linux/* wird dabei festgelegt, welche Treiber (und damit welche Hardware) der neue Kernel unterstützen soll.
5. Das Kommando *make dep* sollte ausgeführt werden, damit sichergestellt ist, daß alle Include Dateien an der richtigen Stelle gespeichert sind.
6. Das Kommando *make clean* stellt sicher, daß alle Teile des Kernels neu kompiliert werden. D.h. Object Files älterer Versionen werden gelöscht.
7. Das Kompilieren wird in den meisten Fällen mit *make zImage* gestartet (Ausnahme: Erstellen einer Bootdiskette). Je nach verwendetem Prozessor (und bereitstehenden Arbeitsspeicher) dauert dies zwischen 10 Minuten und 2 Stunden.

4.2 Mobile-IP Erweiterungen

Um die Implementierung von Mobile-IP für Linux der University of Singapore zu verwenden, ist es notwendig einige Veränderungen am Kernel vorzunehmen. Hierzu stehen die nötigen Ergänzungen für den Kernel-Quellcode auf dem FTP-Server[SING] der University of Singapore bereit.

Die beiden Dateien

- mip-daemon-1.0.tar.gz
- mip-kernel-1.0.tar.gz

sollten im Verzeichnis */usr/src* entpackt werden.

4.2.1 Installation der Kernel-Erweiterungen

Die Ergänzungen für den Linux Kernel befinden sich nach dem Entpacken im Verzeichnis */usr/src/mip-kernel-1.0/*. Mit dem Befehl

```
patch -p0 < /usr/src/mip-kernel/mip-patch-1.0
```

im Verzeichnis */usr/src* ausgeführt, werden die Ergänzungen am Kernel-Quellcode vorgenommen.

Anschließend ist es notwendig, den Kernel neu zu kompilieren.

Während *make config* ist dann anzugeben, ob der Rechner als MN, als FA oder aber als HA verwendet werden soll[6].

4.2.2 Installation der Dämonprogramme

Im Verzeichnis */usr/src/mip-daemon-1.0/* befinden sich Programme, welche als Hintergrundprozesse während ein Rechner als Agent oder Node arbeitet laufen müssen. Diese werden mit dem Kommando *make all* kompiliert und mit *make install* in die entsprechenden Verzeichnisse für ausführbare Dateien bzw. für Konfigurationsdateien kopiert.

[6] Ein Rechner kann auch gleichzeitig als HA und FA verwendet werden

4.2.3 Konfigurationsdateien

Die beiden Dateien */etc/mnserv.conf* und */etc/agtserv.conf* müssen auf allen beteiligten Rechnern an das eigene System angepaßt werden.

Konfigurationsdatei */etc/agtserv.conf* **für einen HA.**

Wird ein Rechner als HA verwendet, so ist die Datei */etc/agtserv.conf* wie folgt zu modifizieren:

```
# configure global variable of mobile IP stack

global {
unimode     yes # HA: cannot accept simutaneous registration
autoansr    yes # HA: answer REGISTRATION REQUEST automatically
autofwd     yes # FA: forwarding REGISTRATION REQUEST automatically
regAwaitFA  20 # FA: registration awaiting time
regOffsetFA 1 # FA: registration lifetime offset
regAwaitHA  10 # HA: registration awaiting time
regOffsetFA 1 # HA: registration lifetime offset
regLifetime 120 # HA: default granted registration lifetime
}
```

Diese Default-Einstellungen können beibehalten werden.

```
# wireless interface

device {
        dev 129.187.214.42  adv yes  advIntvl 2  lifetime 72000 FA_AND_HA
# the interface sends advertisement every 2 seconds
# the interface can offer FA service for 20 hours
# you can set the agent type to FA, HA or FA_AND_HA

}
```

Hier wird das Interface spezifiziert, über welches der HA Advertisements verschickt, bzw. ueber das er den Verkehr mit dem FA abwickelt.

```
# wireless router address within the same subnet

router {
        router 129.187.214.254  prefer 9000
router 129.187.214.48   prefer 9000
# can anybody tell how to configure preference level ?
}
```

Einzutragen ist die Adresse, der für das Subnetz, in dem der Agent betrieben wird, verwendeten Router.

```
# care-of addresses, usually those wired interfaces
# required only for Foreign Agent

coaddr {
192.168.214.109
```

```
}
```

Da die Care-of Adresse durch das Ende des Tunnels gegeben wird, ist hier im Normalfall die Adresse des FA einzutragen.

```
key {
addr 129.187.214.44  spi 666  543   key 044^042
}
```

In [RFC1321] wird das MD5 Verfahren spezifiziert. Dieses kommt hier zur Anwendung.

Konfigurationsdatei */etc/agtserv.conf* für einen FA

Wird ein Rechner als FA verwendet, so ist die Datei */etc/agtserv.conf* wie folgt zu modifizieren:

```
# configure global variable of mobile IP stack

global {
unimode     yes # HA: cannot accept simutaneous registration
autoansr    yes # HA: answer REGISTRATION REQUEST automatically
autofwd     yes # FA: forwarding REGISTRATION REQUEST automatically
regAwaitFA  20  # FA: registration awaiting time
regOffsetFA 1   # FA: registration lifetime offset
regAwaitHA  10  # HA: registration awaiting time
regOffsetFA 1   # HA: registration lifetime offset
regLifetime 120 # HA: default granted registration lifetime
}
```

Diese Default-Einstellungen können beibehalten werden.

```
# wireless interface

device {
dev 192.168.214.109  adv yes  advIntvl 2  lifetime 72000 FA_AND_HA
# the interface sends advertisement every 2 seconds
# the interface can offer FA service for 20 hours
# you can set the agent type to FA, HA or FA_AND_HA

}
```

Hier wird das Interface spezifiziert, über welches der FA Advertisements verschickt, bzw. ueber das er den Verkehr mit dem FA und dem MN abwickelt.

```
# wireless router address within the same subnet

router {
router 192.168.214.254  prefer 9000
# can anybody tell how to configure preference level ?
}
```

Auch hier müssen, wie oben, die Adressen der für das Subnetz zuständigen Router eingetragen werden.

```
# care-of addresses, usually those wired interfaces
# required only for Foreign Agent

coaddr {
192.168.214.109
}
```

Wenn ein Rechner als FA betrieben wird, so kann dieser Abschnitt ignoriert werden.

```
key {
addr 129.187.214.44  spi 666  543   key 044^042
}
```

In [RFC1321] wird das MD5 Verfahren spezifiziert. Dieses kommt hier zur Anwendung. Zu den Einstellungen, siehe im folgenden Beispiel.

Konfigurationsdatei */etc/mnserv.conf* **für einen MN:**

Soll ein Rechner als MN verwendet werden, so ist die Datei */etc/mnserv.conf* wie folgt zu bearbeiten:

```
# wireless interface of mobile node

device {
dev 129.187.214.44  solIntvl 3  regIntvl 10  regOffset 1  maxSols 10
# the interface sends Agent Solicitation not more often than every 3 seconds
# registration retransmission should not be more often than every 10 seconds
# re-registration should be 1 second earlier before lifetime is finished
}
```

Ebenso wie bei den Agenten, ist auch hier das Device einzutragen, mit dem der Node mit den Agenten kommuniziert. Auch hier dürfte es sich also im Normalfall um die eigene IP-Adresse handeln.

```
# registration issued by daemon
# able to regitser more than one home agent by adding entries with various home agents

registration {
home_agent 129.187.214.42 lifetime 600  mode remove spi_mnha 555 243  spi_mnfa 666 543
}
```

Die IP Adresse des als HA vorgesehenen Rechners wird in dieser Section eingetragen.

```
key {
addr 129.187.214.42  spi 666  543   key 044^042
```

```
addr 192.168.214.109 spi 555  241   key 044^042
}
```

Das in [RFC1321] spezifizierte Verfahren wird hier angewendet.

Zu beiden Agenten muß jeweils ein Eintrag existieren. Die hinter key eingetragenen Werte müßen gleich sein.

Bsp.: HA, FA, und MN seien konfiguriert wie in Abbildung 3.

Dann können die folgenden Einträge in den jeweiligen .conf Dateien eingetragen werden.

```
HA: addr 129.187.214.44  spi 666  543   key 044^042   (Datei: agtserv.conf)
FA: addr 129.187.214.44  spi 555  243   key 044^042   (Datei: agtserv.conf)
MN: addr 129.187.214.42  spi 666  543   key 044^042   (Datei: mnserv.conf)
    addr 192.168.214.109 spi 555  241   key 044^042

# This item may be a duplicate the file /etc/rc.d/rc.inet1. Hopefully, this can be
# fixed in future.
home_net {
dev 129.187.214.44  net 129.187.214.0  netmask 255.255.255.0  gw 129.187.214.44
dev 129.187.214.44  net 0.0.0.0        netmask 0.0.0.0        gw 129.187.214.254
}
```

Diese Sektion soll den MN wieder über sein heimatliches Subnetz informieren. D.h. man benötigt i.a. zwei Zeilen[7].

dev x.x.x.x spezifiziert die Ethernetkarte

net x.x.x.x spezifiziert das Netz (z.B. 129.187.214.0)

netmask x.x.x.x spezifiziert die Netmask

gw x.x.x.x spezifiziert das zu verwendende Gateway.

Wobei in einer Zeile das heimatliche Subnetz und in einer weiteren Zeile „der Rest des Internets“ beschrieben wird.

Anm.: in der Zeile, die das heimatliche Subnetz beschreibt, ist als Gateway die IP Adresse des MN einzutragen.

[7] Ausnahme: der MN soll im heimatlichen Subnetz als Router arbeiten

4.2.4 Mobile-IP im Betrieb

Um Mobile-IP zu verwenden, müßen die oben erwähnten Dämon Programme gestartet werden. Ist ein Rechner als Agent vorgesehen, so startet man die entsprechenden Prozesse mit *mip start* bzw. bei einem MN durch den Befehl *nip start*[8].

Nun ist es möglich, nach belieben, den MN in den beiden Subnetzen anzuschließen; der MN ist immer mit der gleichen IP-Adresse erreichbar, bzw. er kann auch ganz normal am Netzverkehr teilnehmen.

Im folgenden wird der Betrieb von Mobile-IP analysiert:

Ping von einem Rechner des Sun-Clusters am LRZ aus zum MN:

MN befindet sich im heimatlichen Subnetz:

```
sun1-55: ping -I 1 pchegering4.nm.informatik.uni-muenchen.de 64 5
PING pchegering4.nm.informatik.uni-muenchen.de: 64 data bytes
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=0. time=17. ms
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=1. time=8. ms
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=2. time=4. ms
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=3. time=7. ms
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=4. time=5. ms

----pchegering4.nm.informatik.uni-muenchen.de PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 4/8/17
```

MN befindet sich im fremden Subnetz:

```
sun1-58: ping -I 1 pchegering4.nm.informatik.uni-muenchen.de 64 5
PING pchegering4.nm.informatik.uni-muenchen.de: 64 data bytes
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=0. time=9. ms
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=1. time=8. ms
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=2. time=8. ms
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=3. time=9. ms
72 bytes from pchegering4.nm.informatik.uni-muenchen.de (129.187.214.44): icmp_seq=4. time=8. ms

----pchegering4.nm.informatik.uni-muenchen.de PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 8/8/9
```

Dieses etwas unerwartete Ergebnis kann dadurch erklärt werden, daß sich zischen dem 129.x.x.x und dem 192.x.x.x Netz nur ein Router befindet.

[8]Während der Installation wurden auch entsprechende Manual Pages, welche mit man mip, bzw. man nip, aufgerufen werden, installiert

Telnet-Verbindungen:

Es ist möglich vom MN aus, bzw. zum MN, eine telnet-Verbindung aufzubauen, jedoch wird diese nicht aufrecht erhalten, wenn der MN seinen Ort wechselt.

5 Zusammenfassung/Ausblick

Abschließend sei über die vorliegende Software noch zu erwähnen, daß es durchaus als Schritt in die richtige Richtung zu betrachten ist. Jedoch an manchen Stellen noch einige Schwächen aufweist.

Neben der mangelhaften Dokumentation (wireless interface,...) ist in meinen Augen vor allem noch Inpu/Output zu verbessern:

An vernünftiges Arbeiten ist nicht zu denken, wenn Registrierungsmeldungen einfach in den Bildschirm geschrienben werden.

Literatur

[CHAP95] Graham Chapman: The Linux Bootdisk HOWTO

ftp://ftp.leo.org/pub/comp/os/linux/slackware/docs/Bootdisk-HOWTO

[KIRC95] Olaf Kirch: Network Administrator Guid. O'Reilly & Associates, INC (1995).

[KOFL95] Michael Kofler: Linux: Installation, Konfiguration, Anwendung. Addison-Wesley (1995)

[LEO] LEO Software Archiv: Link everything online,

im HTML Format: http://www.leo.org

IP-Adr.: 131.159.0.198

[NYSTATE] WWW Seiten zur Implementierung von Mobile-IP Birmingham Version:

im HTML Format: http://anchor.cs.binghamton.edu/ mobileip

IP-Adr.: 128.226.3.28

[PERK95] Charles Perkins: IP Encapsulation within IP. Internet Draft – in Bearbeitung, Oktober 1995.

[PERK96] Charls Perkins. draft-ietf-mobileip-16.txt – in Bearbeitung, April 1996.

[RFC1180] T. Socolofsky,C. Kale: A TCP/IP Tutorial. RFC 1180, Januar 1991.

[RFC1256] Stephen E. Deering: ICMP Router Discovery Messages. RFC 1256, September 1991.

[RFC1321] Ronald L. Rivest: The MD5 Message-Digest Algorithm. RFC 1321, April 1992

[SING] FTP Server der National University of Singapore: ftp://ftp.nus.sg/pub/mobile-ip

[WARD95] Brian Ward: The Linux Kernel HOWTO. Graz (1995)

ftp://ftp.leo.org/pub/comp/os/linux/slackware/docs/Kernel-HOWTO,

[WEKA95] Matt Welsh, Lars Kaufman: Running Linux, O'Reilly & Associates, INC (1995).