Institut für Informatik
der Ludwig-Maximilians-Universität
München

Weiterentwicklung der Testumgebung für die Erprobung von Protokollen zur Unterstützung mobiler Systeme

Fortgeschrittenenpraktikum am Lehrstuhl für Rechnernetze und Systemprogrammierung

Bearbeiter: Robert Hoffmann
Betreuer: Stephen Heilbronner
Aufgabensteller: Prof. Dr. H.-G. Hegering

6. Dezember 1996

Inhaltsverzeichnis

1 Einleitung **3**

2 Die Testumgebungen **4**

2.1 Interne Testumgebung 4

2.2 Die Bristol-, Rennes- und Singapur-Testumgebung 9

3 Die Durchführung von Tests **14**

3.1 Tests in der internen Testumgebung 14

3.2 Tests mit Bristol 18

3.3 Tests mit Rennes 19

3.4 Tests mit Singapur 20

4 Zusammenfassung und Ausblick **21**

Abbildungsverzeichnis

1 Interne Testumgebung 5
2 Bristol-, Rennes- und Singapur- Testumgebung 13

1 Einleitung

Die Aufgabe dieses Fortgeschrittenenpraktikums bestand darin, eine bestehende Testumgebung für die Erprobung von Protokollen zur Unterstützung mobiler Systeme weiterzupflegen, Tests durchzuführen und die dazu notwendige Konfigurierung der Rechner vorzunehmen.

Getestet wurde die Funktionsfähigkeit der Mobile-IP Implementation der National University of Singapore in der internen Testumgebung des Instituts sowie unter realen Bedingungen mit den Testpartnern

- Stephanie Ginguene und Sylvain Gombault am ENSTB - Departement Reseaux et Systemes Multimedia in Rennes, Frankreich,
- Manuel Rodriguez an den Hewlett Packard Laboratories, Bristol, Großbritannien und
- Jiang Ming Liang und dem Autor der Singapur-Mobile-IP-Software Yunzhou Li an der National University of Singapore.

Bei den Tests mit der Testumgebung in Bristol war außerdem die Interoperabilität der Singapur-Implementation mit der HP-eigenen Mobile-IP-Implementation, die in Bristol verwendet wird, von Interesse.

2 Die Testumgebungen

Die interne Testumgebung war in einem vorangegangenen Fortgeschrittenenpraktikum (siehe [LAN96]) eingerichtet worden. Sie bestand zu Beginn dieses Praktikums aus drei PCs und einem Notebook, auf denen das Betriebssystem Linux in der Version 1.3.59 mit der Erweiterung zur Unterstützung von Mobile-IP der Universität von Singapur in der Version 1.0 installiert war.

Die im Laufe des Praktikums veröffentlichten Versionen 1.1 und 1.2 dieser Mobile-IP Erweiterung wurden auf den Rechnern der Testumgebung installiert. Auf die Installationsprozeduren der diese Erweiterung umfassenden Kernelpatches und Daemonprogramme wird hier nicht näher eingegangen. Dies ist in [LAN96] und der dem Mobile-IP Paket beigefügten README-Datei ausführlich dokumentiert.

Einer der PCs diente ursprünglich nur als Router für ein privates, ein „fremdes Netz" simulierendes Subnetz. Im Rahmen der Tests mit den Testpartnern in Bristol, Rennes und Singapur wurde ein weiterer ständig verfügbarer Rechner benötigt, der die Rolle eines Mobile Node im fremden Netz übernehmen konnte. Dieser PC wurde deshalb mit Skripten ausgestattet, die ein einfaches Umschalten seines Namens, seiner IP-Adresse, Routingtabelle und Mobile-IP Konfiguration erlauben.

Der Rechner, der im internen Testbetrieb als Home Agent fungiert, wurde so eingerichtet, daß er ohne vorherige Änderung im Test mit den Testpartnern auch als Foreign Agent für den zum Mobile Node umgeschalteten Router dienen kann.

2.1 Interne Testumgebung

Die institutsinterne Testumgebung besteht aus dem Home Agent (HA) *pchegering2*, dem Foreign Agent (FA) *pchegering9* im privaten Subnetz 192.168.214.0 (siehe [RFC1597]), das über den Router *pchegering8* mit dem Institutsnetz verbunden ist. Wird der Mobile Node (MN) *pchegering4* an Hub 1 angeschlossen, befindet er sich in seinem Heimatnetz, wird er an Hub 2 angeschlossen, im simulierten fremden Netz. Abbildung 1 zeigt den Aufbau der Testumgebung.

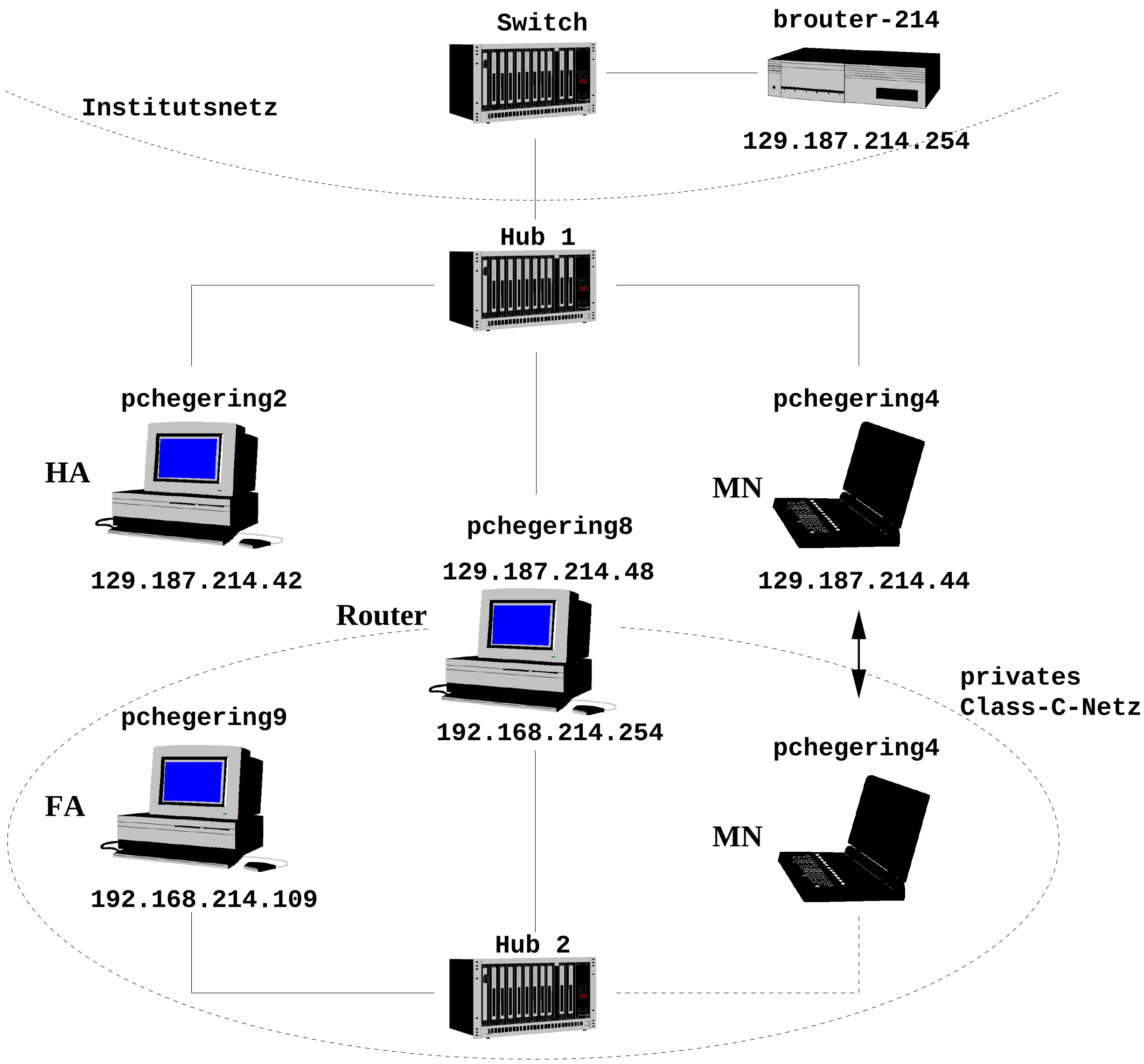

Abbildung 1: Interne Testumgebung

Mit den Dateien `/etc/agtserv.conf` bzw. `/etc/mnserv.conf` sind die Mobile-IP Daemons auf Home- und Foreign Agent bzw. Mobile Node zu konfigurieren. Man kann dabei größtenteils die bereits vorgegebenen Einstellungen beibehalten und muß nur die IP-Adressen der beteiligten Rechner und ggf. die zur Authentifizierung nach dem MD5-Verfahren (siehe [RFC1321]) erforderlichen Schlüssel eintragen.

Die `/etc/agtserv.conf` des HA sieht so aus:[1]

```
# 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      5       # FA: registration awaiting time
  regOffsetFA     1       # FA: registration lifetime offset
  regAwaitHA      2       # HA: registration awaiting time
  regOffsetFA     1       # HA: registration lifetime offset
  regLifetime     120     # HA: default granted registration lifetime
  tunnelFA        yes     # FA: support for bidirectional tunneling
  tunnelHA        yes     # HA: support for bidirectional tunneling
}
```

Die letzten beiden Zeilen sind in Version 1.2 hinzugekommen, in der nun auch bidirektionales Tunneling unterstützt wird. Mehr dazu in Abschnitt 3.1.

```
# wireless interface

device {
  dev 129.187.214.42  adv yes  advIntvl 1  lifetime 36000 FA_AND_HA
  # the interface sends advertisement every 1 second
  # the interface can offer FA service for 10 hours
  # you can set the agent type to FA, HA or FA_AND_HA
}

# wireless router address within the same subnet
```

[1] Kommentare zum Teil gelöscht

```
router {
  router 129.187.214.42  prefer 9000
}
```

Der Grund, weshalb hier statt der Adresse des üblichen *brouter-214* (129.187.214.254) die eigene Adresse angegeben wird, ist in Abschnitt 3.2 beschrieben.

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

coaddr {
  129.187.214.42
}
```

pchegering2 ist zugleich HA und FA[2]. Wird der Rechner nur als HA verwendet, darf keine Care-of Adresse eingetragen werden.

```
key {
  # eigenes Testbett
  addr 129.187.214.44  spi 666  543   key testkey_for_home

  # Bristol
  addr 129.187.214.71  spi 666  543   key testkeyf_bristol

  # Rennes
  addr 129.187.214.72  spi 666  543   key 072^042

  # Thomas Lopatic
  addr 129.187.214.73  spi 666  543   key 073^042
}
```

Die beiden Zahlen hinter `spi` („Security Parameter Index“) sind beliebig und werden ignoriert. Zur Authentifizierung wird nur die maximal 16 Zeichen lange Zeichenkette hinter `key` verwendet.

[2]für die Bristol-, Rennes- und Singapur-Testumgebung, siehe Abschnitt 2.2

Die /etc/agtserv.conf des FA unterscheidet sich von der des HA im wesentlichen in den IP-Adressen:

```
# 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      5       # FA: registration awaiting time
  regOffsetFA     1       # FA: registration lifetime offset
  regAwaitHA      2       # HA: registration awaiting time
  regOffsetFA     1       # HA: registration lifetime offset
  regLifetime     120     # HA: default granted registration lifetime
  tunnelFA        yes     # FA: support for bidirectional tunneling
  tunnelHA        yes     # HA: support for bidirectional tunneling
}

# wireless interface

device {
  dev 192.168.214.109  adv yes  advIntvl 1  lifetime 36000 FA
  # the interface sends advertisement every 1 second
  # the interface can offer FA service for 20 hours
  # you can set the agent type to FA, HA or FA_AND_HA
}

# wireless router address within the same subnet

router {
  router 192.168.214.254 prefer 9000
}

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

coaddr {
  192.168.214.109
```

```
}

key {
}
```

Hier darf kein Schlüssel angegeben werden, andernfalls geschehen rätselhafte Dinge.

`/etc/mnserv.conf` des MN:

```
device {
  dev 129.187.214.44  solIntvl 3  regIntvl 5  regOffset 1  maxSols 10
}

registration {
  home_agent 129.187.214.42 lifetime 600  mode remove \
    spi_mnha 555 243 spi_mnfa 666 543  bi-tunnel yes
}
```

In der vorangegangenen Zeile läßt sich das bidirektionale Tunneling ein- und ausschalten. Die Werte hinter `spi_mnha`, `spi_mnfa` und, in der folgenden Zeile, `spi` werden wie beim HA ignoriert.

```
key {
  addr 129.187.214.42   spi 666  543   key testkey_for_home
}

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
}
```

2.2 Die Bristol-, Rennes- und Singapur-Testumgebung

Um einen ständig verfügbaren Mobile Node wahlweise für Tests mit den Testpartnern in Bristol, Rennes oder Singapur zur Verfügung zu haben, wurde der Rechner *pchegering8*, der in der internen Testumgebung als Router zwischen Instituts- und privatem Subnetz dient, so eingerichtet, daß man auf einfache Weise seine Netz- und Mobile-IP-Konfiguration ändern kann.

Dies geschieht durch Aufruf der Skripten `/etc/make_mn-bristol`, `/etc/make_mn-rennes` bzw. `/etc/make_mn-singapur` und anschließendem Reboot. Die ursprüngliche Konfiguration als *pchegering8* erhält man durch Ausführen von `make_pchegering8`.

`/etc/make_mn-bristol` enthält die folgenden Befehle:

```
cp /etc/resolv.conf.mn-bristol /etc/resolv.conf
cp /etc/HOSTNAME.mn-bristol /etc/HOSTNAME
cp /etc/rc.d/rc.inet1.mn-bristol /etc/rc.d/rc.inet1
cp /etc/fstab.mn-bristol /etc/fstab
cp /etc/mnserv.conf.mn-bristol /etc/mnserv.conf
cp /etc/exports.mn-bristol /etc/exports
```

Die übrigen Skripten funktionieren analog dazu.

Die `/etc/fstab.mn*` enthalten keine von *pchegering2* zu mountenden NFS-Dateisysteme, `/etc/exports.mn*` sind leer, in `/etc/resolv.conf.mn*` und `/etc/HOSTNAME.mn*` sind die Domain- bzw. Rechnernamen entsprechend geändert und `/etc/rc.d/rc.inet1.mn*` stellen die IP-Adressen und Routingtabellen ein.

Die MIP-Konfigurationsdateien für Version 1.2 sehen so aus:

`/etc/mnserv.conf.mn-bristol`:

```
device {
  dev 194.73.207.205  solIntvl 3  regIntvl 5  regOffset 1  maxSols 10
}

registration {
  home_agent 194.73.207.201 lifetime 600  mode remove spi_mnha 555 243 \
    spi_mnfa 666 543  bi-tunnel yes
}

key {
  addr 194.73.207.201     spi 666 543 key GermanyBristolkey
}
```

```
home_net {
  dev 194.73.207.205  net 194.73.207.0  netmask 255.255.255.0 \
    gw 194.73.207.205
  dev 194.73.207.205  net 0.0.0.0        netmask 0.0.0.0      \
    gw 194.73.207.1
}
```

/etc/mnserv.conf.mn-rennes:

```
device {
  dev 193.52.74.111  solIntvl 3  regIntvl 5  regOffset 1  maxSols 10
}

registration {
  home_agent 193.52.74.110 lifetime 600  mode remove spi_mnha 555 243 \
    spi_mnfa 666 543  bi-tunnel yes
}

key {
  addr 193.52.74.110      spi 555 243 key 110^111
}

home_net {
  dev 193.52.74.111  net 193.52.74.64  netmask 255.255.255.192 \
    gw 193.52.74.111
  dev 193.52.74.111  net 0.0.0.0        netmask 0.0.0.0        \
    gw 193.52.74.66
}
```

/etc/mnserv.conf.mn-singapur:

```
device {
  dev 137.132.153.251  solIntvl 3  regIntvl 5  regOffset 1  maxSols 10
}

registration {
  home_agent 137.132.153.225 lifetime 600  mode remove spi_mnha 555 243 \
    spi_mnfa 666 543  bi-tunnel yes
}
```

```
key {
  addr 137.132.153.225    key robert
}

home_net {
  dev 137.132.153.251  net 137.132.153.224 netmask 255.255.255.240 \
    gw 137.132.153.251
  dev 137.132.153.251  net 0.0.0.0        netmask 0.0.0.0           \
    gw 137.132.153.225
}
```

pchegering2, der in beiden Testumgebungen als Home Agent fungiert, wurde bereits die zusätzliche Funktion eines Foreign Agents für den zum Mobile Node umkonfigurierten *pchegering8* gegeben (siehe Abschnitt 2.1). Beim einem Wechsel zwischen den Testumgebungen muß daher an seiner Konfiguration nichts geändert werden.

Abbildung 2 zeigt die komplette Testumgebung.

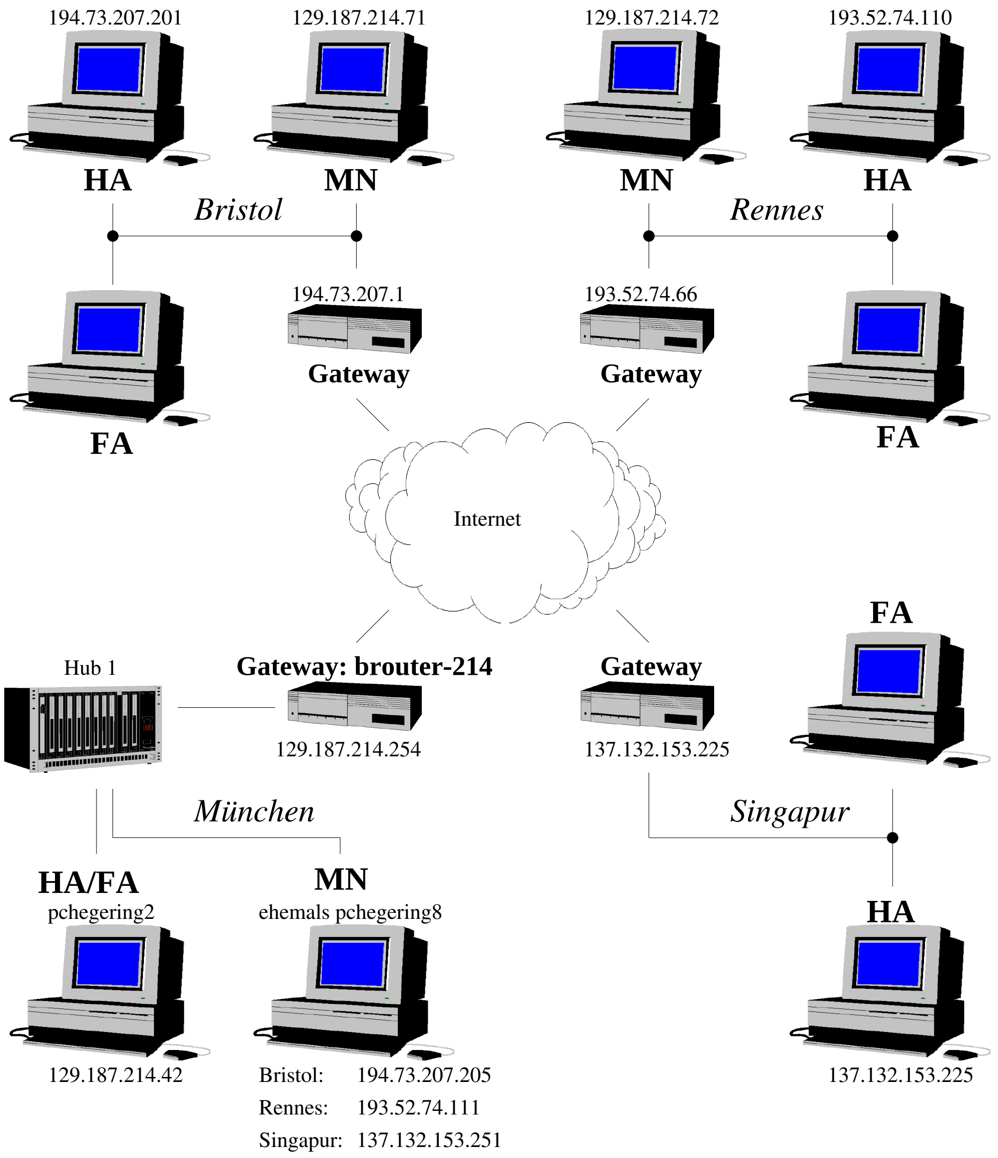

Abbildung 2: Bristol-, Rennes- und Singapur- Testumgebung

3 Die Durchführung von Tests

Die MIP-Versionen 1.0, 1.1 und 1.2 wurden über einen Zeitraum von drei Monaten getestet. War eine neu veröffentlichte Version installiert, wurde sie zunächst intern getestet, später dann zusammen mit den Testpartnern in Bristol und Rennes, mit den Partnern in Singapur nur Version 1.2.

Dabei auftretende Probleme und Lösungsvorschläge wurden über eine Mailingliste diskutiert. Einige dieser Vorschläge flossen in spätere MIP-Implementierungen ein und beseitigten manche der Probleme.

Die Ursachen waren zum Teil in Fehlern der Implementierung begründet, zum Teil waren sie netztopologischer Natur. Erstere wurden von den Entwicklern bis zur Version 1.2 größtenteils korrigiert, letztere muß man jedoch als gegeben betrachten und an anderer Stelle nach Lösungen suchen. So lassen sich beispielsweise bestimmte Sicherheitsmechanismen von Routern nicht ohne weiteres abschalten.

Ein erhebliches Problem bei den Tests mit den Partnern im Ausland war die geringe Bandbreite der Verbindungen. Vor allem die Verbindung nach Singapur war fast unbrauchbar.

3.1 Tests in der internen Testumgebung

Bei den ersten Versuchen, mit MIP-Version 1.0 die eigene Testumgebung in Betrieb zu nehmen, fielen vor allem Implementierungsfehler auf. So funktionierte die Registrierung des Mobile Node im fremden Netz nur dann, wenn auf dem MN gleichzeitig ein `tcpdump`-Prozeß lief, der das Netzinterface in den „promiscous mode“ versetzte und damit für Ethernet-Pakete, die an andere Interfaces adressiert waren, empfänglich machte. Der promiscous mode läßt sich auch mit `ifconfig eth0 promisc` einschalten. Dies war in den Folgeversionen nicht mehr notwendig.

Applikationsverbindungen wie `telnet` wurden beim Umstecken des MN vom Heimatnetz ins fremde Netz oder umgekehrt abgebrochen. Beim Umstecken vom fremden Netz ins Heimatnetz mußte zusätzlich der MIP-Daemon des MN neu gestartet werden (`nip restart`), da der Daemon den Routingeintrag zum Foreign Agent nicht löschte und auch die ursprüngliche Defaultroute nicht wieder herstellte. Dies trat ab Version 1.1 nicht mehr auf.

In MIP-Version 1.2 hat Thomas Lopatic (Uni München) die Option „bidirektionales Tunneling“ nach [MON96] eingebaut. Sie soll dafür sorgen, daß IP-Pakete, die vom Mobile Node kommen, denselben Weg über einen Tunnel zwischen Foreign Agent und Home Agent nehmen, wie Pakete, die an den MN gesendet werden. In [PER96], das der verwendeten MIP-Implementierung zu Grunde liegt, ist dagegen vorgesehen, daß der MN Pakete auf normalem Wege ohne Tunnel versendet.

Beim Betrieb in der internen Testumgebung hat dies keine praktische Bedeutung, im Gegensatz zum Betrieb im realen Internet, wie die Tests mit den Partnern in Bristol zeigten (siehe Abschnitt 3.2). Ein `ping` vom MN auf beliebige andere Rechner innerhalb oder außerhalb des Universitätsnetzes außer dem HA funktionierte, egal ob „bidirektionales Tunneling“ eingeschaltet war oder nicht. Allerdings scheinen in der Implementierung noch Fehler vorzuliegen, denn die Antwortpakete wurden meistens am FA dupliziert. Ein `ping` vom MN auf *zeus.cip* ergibt:

```
PING 129.187.214.129 (129.187.214.129): 56 data bytes
64 bytes from 129.187.214.129: icmp_seq=0 ttl=253 time=13.3 ms
64 bytes from 129.187.214.129: icmp_seq=0 ttl=253 time=16.8 ms (DUP!)
64 bytes from 129.187.214.129: icmp_seq=1 ttl=253 time=19.0 ms
64 bytes from 129.187.214.129: icmp_seq=1 ttl=253 time=20.6 ms (DUP!)
64 bytes from 129.187.214.129: icmp_seq=2 ttl=253 time=19.0 ms
64 bytes from 129.187.214.129: icmp_seq=2 ttl=253 time=20.7 ms (DUP!)
64 bytes from 129.187.214.129: icmp_seq=3 ttl=253 time=22.0 ms
64 bytes from 129.187.214.129: icmp_seq=3 ttl=253 time=23.5 ms (DUP!)
64 bytes from 129.187.214.129: icmp_seq=4 ttl=253 time=11.8 ms
64 bytes from 129.187.214.129: icmp_seq=4 ttl=253 time=22.6 ms (DUP!)

--- 129.187.214.129 ping statistics ---
5 packets transmitted, 5 packets received, +5 duplicates, 0% packet loss
```

Ein `tcpdump` auf dem FA zeigt:

```
10:18:40.738687 pchegering4.nm.informatik.uni-muenchen.de > \
  zeus.cip.informatik.uni-muenchen.de: icmp: echo request
10:18:40.748687 zeus.cip.informatik.uni-muenchen.de > \
  pchegering4.nm.informatik.uni-muenchen.de: icmp: echo reply
```

```
10:18:40.748687 zeus.cip.informatik.uni-muenchen.de > \
  pchegering4.nm.informatik.uni-muenchen.de: icmp: echo reply
```

Ein weiterer Fehler wurde mit Version 1.2 offenbar neu eingeführt: Verbindungen zwischen MN und HA kamen in den Versionen 1.0 und 1.1 zustande, in Version 1.2 nicht. Ein `ping`-Test vom HA zum MN ergab, nur das erste Paket wurde beantwortet, bei aktivem bidirektionalem Tunnel dupliziert, danach kamen keine Antworten mehr an, bis die Registrierung erneuert wurde. Die Registrierung kam aber nicht zustande, wenn der ping nicht abgebrochen wurde.

Hier die Registrierungsmeldungen des HA:

```
pchegering2(robert):~ % cat /proc/net/mip_dev
Address         FA  HA  Busy Adv-interval Lifetime NextSeqno
129.187.214.42  Yes Yes No              1      255       231

pchegering2(robert):~ % cat /proc/net/mip_reg
Type UsrRef Regno State Home addr      HA addr        Care-of addr    \\
HA   1      10    4     129.187.214.44 129.187.214.42 192.168.214.109 \\

Registration ID   Lifetime MNport RCT36 Bcast Bi-Tunnel
00000015-B81AFF31      120    434     2 No    Yes
```

Das erste, duplizierte Paket:

```
pchegering2:~# ping 129.187.214.44
PING 129.187.214.44 (129.187.214.44): 56 data bytes
64 bytes from 129.187.214.44: icmp_seq=0 ttl=63 time=13.4 ms
64 bytes from 129.187.214.44: icmp_seq=0 ttl=63 time=14.5 ms (DUP!)
```

Auf dem umgekehrten Weg, einem `ping` vom MN zum HA, zeigt `tcpdump` auf dem HA duplizierte Antwortpakete:[3]

```
11:46:40.346313 pchegering4.nm.informatik.uni-muenchen.de > \
  pchegering2.nm.informatik.uni-muenchen.de: icmp: echo request
11:46:40.346313 pchegering2.nm.informatik.uni-muenchen.de > \
  pchegering4.nm.informatik.uni-muenchen.de: icmp: echo reply
11:46:41.206313 pchegering2.nm.informatik.uni-muenchen.de > \
```

[3] Die Uhrzeiten sind auf den PCs nicht immer richtig eingestellt.

```
  255.255.255.255: icmp: type-#9 [tos 0x10] [ttl 1]
11:46:41.346313 pchegering4.nm.informatik.uni-muenchen.de > \
  pchegering2.nm.informatik.uni-muenchen.de: icmp: echo request
11:46:41.346313 pchegering2.nm.informatik.uni-muenchen.de > \
  pchegering4.nm.informatik.uni-muenchen.de: icmp: echo reply
11:46:41.346313 pchegering2.nm.informatik.uni-muenchen.de > \
  pchegering4.nm.informatik.uni-muenchen.de: icmp: echo reply
11:46:42.206313 pchegering2.nm.informatik.uni-muenchen.de > \
  255.255.255.255: icmp: type-#9 [tos 0x10] [ttl 1]
```

die im FA verlorengegangen sind:

```
10:26:34.308687 pchegering9.nm.informatik.uni-muenchen.de > \
  255.255.255.255: icmp: type-#9 [tos 0x10] [ttl 1]
10:26:34.978687 pchegering4.nm.informatik.uni-muenchen.de > \
  pchegering2.nm.informatik.uni-muenchen.de: icmp: echo request
10:26:35.308687 pchegering9.nm.informatik.uni-muenchen.de > \
  255.255.255.255: icmp: type-#9 [tos 0x10] [ttl 1]
10:26:35.978687 pchegering4.nm.informatik.uni-muenchen.de > \
  pchegering2.nm.informatik.uni-muenchen.de: icmp: echo request
```

Ist bidirektionales Tunneling nicht eingeschaltet, wird die Antwort nicht dupliziert:

```
pchegering4:~# ping 129.187.214.42
PING 129.187.214.42 (129.187.214.42): 56 data bytes
64 bytes from 129.187.214.42: icmp_seq=0 ttl=62 time=11.0 ms
```

aber der HA antwortet nicht:

```
21:26:21.223269 pchegering4.nm.informatik.uni-muenchen.de > \
  pchegering2.nm.informatik.uni-muenchen.de: icmp: echo request
21:26:21.793269 pchegering2.nm.informatik.uni-muenchen.de > \
  255.255.255.255: icmp: type-#9 [tos 0x10] [ttl 1]
21:26:22.213269 pchegering4.nm.informatik.uni-muenchen.de > \
  pchegering2.nm.informatik.uni-muenchen.de: icmp: echo request
21:26:22.793269 pchegering2.nm.informatik.uni-muenchen.de > \
  255.255.255.255: icmp: type-#9 [tos 0x10] [ttl 1]
```

Der Autor der MIP-Software, Yunzhou Li, hat bei eigenen Tests keine Beobachtungen dieser Art gemacht. Eine Lösung, evtl. mit der nächsten MIP-

Version, steht noch aus.

3.2 Tests mit Bristol

Zu den Tests mit der HP-eigenen MIP-Implementierung von Manuel Rodriguez in Bristol wurden die MIP-Versionen 1.1 und 1.2 verwendet. In Version 1.0 waren einige Bits der Flags für Advertisements und Registration Requests falsch angeordnet, so daß eine Verständigung der beiden Implementierungen nicht möglich war.

Nach Korrektur der Bitreihenfolge in Version 1.1 glückten die Registrierungen des als Mobile Node *mip-gr.bristol.glocal.net* (194.73.207.205) umkonfigurierten *pchegering8* im Institutsnetz mit seinem Home Agent 194.73.207.201 in Bristol und des MN 129.187.214.71 im Netz von Bristol mit seinem HA *pchegering2* (129.187.214.42).

Allerdings war *pchegering2*, der gleichzeitig als Foreign Agent diente, der einzige Rechner, zu dem der MN *mip-gr* eine Verbindung herstellen konnte. Hier spielte der *brouter-214* bzw. der Router ins WAN hinter diesem nicht mit, jedoch aus einem anderen Grund, als zunächst angenommen.

Router werden nach Auskunft des Leibniz Rechenzentrums gewöhnlich so konfiguriert, daß sie

1. „routing loops“ verhindern, indem sie an Rechner innerhalb des Uni-Netzes adressierte Pakete nicht
   (a) ins Internet hinaus und
   (b) nicht von außen hereinlassen,
2. sowie „spoofing attacks“ verhindern, indem sie keine Pakete mit einer Absenderadresse eines Rechners
   (a) innerhalb des Uni-Netzes herein und
   (b) außerhalb des Uni-Netzes hinauslassen.
3. Außerdem beantworten LRZ-Router keine ARP-Anfragen von Rechnern mit Adressen außerhalb des Uni-Netzes.

Regel 2b war hier nicht die Ursache, da der Router in diesem Punkt nicht das tat, was er sollte, sondern derartige Pakete passieren ließ. Aber Regel 3 veranlaßte den *brouter-214*, seine Hardwareadresse nicht an den Mobile Node zu verraten (sie war mit 00:00:00:00:00:00 im ARP-Cache des MN eingetragen). Dies erklärt auch, warum kein Rechner innerhalb des 214er-Netzes außer dem FA, zu dem Pakete vom MN direkt geroutet werden, erreichbar war.

Diese Eigenschaft des *brouter-214* läßt sich aber dadurch umgehen, daß nicht *brouter-214* als Gateway für den MN in die Konfigurationsdatei `/etc/agtserv.conf` des FA eingetragen wird, sondern der FA selbst.

Da die MIP-Implementierung in Bristol bidirektionales Tunneling nicht unterstützt, konnte Regel 2a nicht umgangen werden, die verhindert, daß der MN 129.187.214.71 in Bristol Verbindung zu einem Rechner innerhalb des Uni-Netzes aufnehmen kann. Dies ist besonders unbefriedigend, da Benutzer Mobiler Rechner in fremden Netzen gerade auch auf Resourcen im Heimatnetz zugreifen wollen. Der Router in Bristol filterte Pakete dieser Art nicht aus.

3.3 Tests mit Rennes

Bei den Tests mit Rennes ergaben sich teilweise nicht erklärbare Schwierigkeiten. Mit MIP-Version 1.1 auf beiden Seiten konnte der als MN *mip-mn-munich.rennes.enst-bretagne.fr* (193.52.74.111) konfigurierte *pchegering8* sich bei seinem HA 193.52.74.110 registrieren, jedoch wurden Registration Requests des MN 129.187.214.72 in Rennes von dessen HA *pchegering2* (129.187.214.42) nicht beantwortet.

Bei MIP-Version 1.2 „verlor“ der MN *mip-mn-munich* regelmäßig seine Registrierung nach einigen Sekunden, egal, ob bidirektionales Tunneling eingeschaltet war, oder nicht. Dasselbe passierte mit dem MN 129.187.214.72 in Rennes, wenn der bidirektionale Tunnel nicht aktiv war. War er eingeschaltet, erreichten seine Registration Requests den HA *pchegering2* nicht.

3.4 Tests mit Singapur

Zu diesen Tests wurde *pchegering8* als MN 137.132.153.251 konfiguriert. Ein MN mit einer IP-Adresse aus dem Uni-Netz wurde in Singapur nicht installiert. Diskussionen fanden mittels `telnet` und `ytalk` während der Tests statt, was wegen der geringen Bandbreite der Verbindung nach Singapur sehr mühsam war. Es ist nicht auszuschließen, daß dies auch Auswirkungen auf das Testergebnis hatte, da sogar die `telnet`-Verbindungen mehrmals abgebrochen wurden.

Das Ergebnis ähnelt dem mit Rennes, der MN „verlor“ seine Registrierung regelmäßig nach wenigen Sekunden. Die Advertisements eines zweiten Home Agents, der zufällig am Datenbank-Lehrstuhl zur gleichen Zeit lief, störten den MN offenbar derart, daß zunächst überhaupt keine Registrierung zustande kam, bis der HA ausgeschaltet war.

4 Zusammenfassung und Ausblick

Die Tests haben die prinzipielle Funktionsfähigkeit der getesteten MIP-Implementierung und ihre Verträglichkeit mit der HP-eigenen Entwicklung gezeigt. Die Robustheit der Software im realen Gebrauch, insbesondere bei geringer Bandbreite, Konfigurationsfehlern und Wechselwirkungen mit anderweitiger Hard- und Software könnte noch verbessert werden. Auch scheinen sich in Version 1.2 noch kleinere Fehler eingeschlichen zu haben.

Ferner hat sich die Notwendigkeit von bei Bedarf einstellbarem bidirektionalen Tunneling bestätigt. Ohne dies ist Mobile IP in Netzen, in denen allgemein empfohlene Sicherheitsvorkehrungen getroffen wurden, nicht möglich.

Literatur

[LAN96] Thomas Lankes: Mobile-IP, Einrichtung einer Testumgebung unter Linux, Fortgeschrittenenpraktikum am Lehrstuhl Hegering, August 1996.

[MON96] G. Montenegro: Bi-directional Tunneling for Mobile IP – work in progress, September 1996.

[PER96] C. Perkins: IP Mobility Support – work in progress, Mai 1996.

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

[RFC1597] Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot: Address Allocation for Private Internets, RFC 1597, März 1994.

[RFC2002] C. Perkins: IP Mobility Support, Oktober 1996.

[RFC2003] C. Perkins: IP Encapsulation within IP, Oktober 1996.

[RFC2004] C. Perkins: Minimal Encapsulation within IP, Oktober 1996.

[RFC2005] J. Solomon: Applicability Statement for IP Mobility Support, Oktober 1996.

[RFC2006] D. Cong, M. Hamlen, C. Perkins, The Definitions of Managed Objects for IP Mobility Support, September 1996.