ZHW

Projektarbeit SS 2001 - PA2 Sna02

Benutzerauthentisierung in grossflächigen Wireless LANs

Betreuender Dozent: Dr. Andreas Steffen
Partnerfirma: FutureLAB AG

21. Mai bis 06. Juli 2001

Studenten: Pascal Chollet, Tanju Erinmez

Inhaltsverzeichnis

Teil I

Einführung

1 Zusammenfassung

Wenn sich ein Benutzer heute über einen Festnetzanschluss ins Internet einwählt, so kann aus der Rufernummer auf die Identität geschlossen und die Zugangsberechtigung anhand eines vorkonfigurierten Kontos verifiziert werden. Wenn nun versucht wird ähnliche Mechanismen auf grossflächige Wireless LANs zu übertragen, so wird man vor das zusätzliche Problem der Mobilität gestellt werden. Es existiert jedoch mit "Cellular IP" ein interessantes Konzept, welches sich zwar der Mobilität angenommen und effektiv gelöst, aber die Authentisierung, die Autorisierung und das Accounting aussenvorgelassen hat.

Die vorliegende Projektarbeit befasst sich in einer ersten Phase mit einer Konzeptanalyse und Konzepterstellung, welches "Cellular IP" in den erwähnten Schwachpunkten untersucht und Lösungsansätze erarbeitet. Es wurden hierzu Anwendungs- und Problemfelder von grossflächigen Wireless LANs untersucht, um daraus einfach realisierbare aber dennoch sichere Lösungen zu entwickeln, welche den Gedanken der Performance aus der ursprünglichen "Cellular IP"-Idee in das neue Konzept hinüberträgt.

In einer zweiten Phase wurde konkret die Paketauthentisierung aus dem erstellten Konzept herausgegriffen, realisiert und in die Referenz-Implementation von "Cellular IP" integriert. Weil diese Referenz-Implementation selbst sehr instabil läuft, wurde in einer frühen Phase des Projekts darauf verzichtet, auf diesem Fundament aufzubauen. Stattdessen wurde eine netfilterbasierende Lösung erarbeitet, deren Hauptmodul wiederverwendbar und in keinster Weise von der instabilen Referenz-Implementation abhängig ist.

Die Benutzerauthentisierung konnte wegen der ausgedehnten Konzepterstellungsphase nicht mehr implementiert werden. Jedoch existiert im Konzept hierzu eine ausführliche Spezifikation.

Winterthur, Freitag 06. Juli 2001

Pascal Chollet Tanju Erinmez

2 Aufgabenstellung

Kommunikationssysteme (KSy)

Projektarbeiten SS 2001 - PA2 Sna02

Benutzerauthentisierung in grossflächigen Wireless LANs

Studierende:

- Pascal Chollet, IT3b
- Tanju Erinmez, IT3b

Partnerfirma:

- FutureLAB AG, Schwalmenackerstr. 4, 8400 Winterthur (http://www.futurelab.ch)

Termine:

- Ausgabe: Montag, 21.05.2001 14:30 - 15:30 im E523
- Abgabe: Freitag, 6.07.2001

Beschreibung:

Wireless LANs werden immer mehr zum Allgemeingut. Interessant sind vor allem grossfläche Netze, die einen ganzen Campus, ein Firmenareal oder sogar eine ganze Stadt lückenlos abdecken. In solchen Netzen können Hunderte oder sogar Tausende von Benutzern gleichzeitig angemeldet sein. Dadurch wird die zuverlässige und korrekte Authentisierung von berechtigten Benutzern zu einem zentralen Thema.

Die Firma Ericsson hat zusammen mit der Columbia University ein spezielles Routing-Protokoll für Wireless LANs entwickelt, das sich "Cellular IP" nennt. Dieses Verfahren verwendet erweiterte ICMP-Meldungen, um die Routen zu den mobilen Teilnehmern dynamisch aufrecht zu erhalten. Im Rahmen einer ZHW Projektarbeit wurde im E-Gebäude ein "Cellular IP" WLAN auf Linux-Basis aufgestellt und erfolgreich in Betrieb genommen.

Als nächster Schritt soll eine Benutzerauthentisierung auf der Basis des bekannten RSA Public Key Verfahrens realisiert werden. Weiter sollen alle gesendeten Routing-Pakete mit Hilfe eines Message Authentication Codes (MAC) und eines geheimen Session Keys gesichert werden. Für die Implementation der RSA Funktionalität kann die GNU Multiprecision (GMP) Library verwendet werden.

Aufgaben:

- Die Authentisierung von mobilen Teilnehmern soll gemäss Absatz 3.5 des IEFT Drafts "draft-ietf-mobileip-cellularip-00.txt" realisiert und im "Cellular IP" Versuchsnetz der ZHW praktisch getestet werden.
- Die RSA Public Keys der mobilen Teilnehmer können direkt in ihrer rohen Form als Modulus und Exponent durch den Gateway verwendet werden und müssen nicht als X.509 Zertifikate abgelegt werden.
- Für die Authentisierung der "Cellular IP Control Packets" soll der HMAC-Algorithmus gemäss RFC 2104 verwendet werden.

Infrastruktur / Tools:

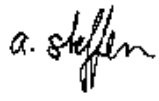
- Raum: **E523**
- Rechner: 7 PCs davon 5 für Wireless Access Nodes, 2 Notebooks
- Hardware: 7 ORiNOCO WLAN Cards, 5 PCMCIA Adapter Cards
- SW-Tools: Linux, "Cellular IP" SW Paket, Linux WLAN Treiber

Literatur / Links:

- ZHW Projektarbeit "Wireless LAN basierend auf Cellular IP"
  http://www.strongsec.com/zhw/PA/PA1_Sna06_2001.pdf
- Cellular IP Home Page
  http://www.comet.columbia.edu/cellularip/
- Columbia IP Micro-Mobility Suite
  http://comet.ctr.columbia.edu/micromobility/
- A. G. Valko, "Cellular IP - A New Approach to Internet Host Mobility"
  http://www.comet.columbia.edu/cellularip/pub/ccr99.pdf
- A. T. Campbell, Gomez, J., Kim, S., Turanyi, Z., Wan, C-Y. and A, Valko "Design, Implementation and Evaluation of Cellular IP"
  http://www.comet.columbia.edu/cellularip/pub/pcs2000.pdf
- IETF Internet Draft "Cellular IP"

  http://www.comet.columbia.edu/cellularip/pub/draft-ietf-mobileip-cellularip-00.txt
- Linux WaveLAN IEEE 802.11 Treiber
  http://www.fasta.fh-dortmund.de/users/andy/wvlan/
- Agere's ORiNOCO Home Page
  http://www.orinocowireless.com
- Wireless LAN resources for Linux
  http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/
- Linux-WLAN Project
  http://www.linux-wlan.com/linux-wlan/
- GMP - GNU Multi-Precision Library
  http://swox.com/gmp/
- IETF RFC 2104
  HMAC: Keyed-Hashing for Message Authentication

Winterthur, 21. März 2001



Dr. Andreas Steffen

3 Einleitung

Die Firma FutureLAB ist mit der Idee an die Zürcher Hochschule Winterthur (ZHW) herangetreten, eine grossflächige Wireless LAN Technologie namens "Cellular IP" anhand einer von der Columbia University erstellten Referenz-Implementation zu untersuchen. Dies wurde in einer vorhergegangen Projektarbeit durchgeführt und hierzu ein Testnetz in der ZHW aufgestellt, welches mit dem ZHW-Schulnetz verbunden ist. Aus den Resultaten jener Projektarbeit lassen sich drei Problemfelder erkennen, die für dieses Projekt von Bedeutung sind:

- Fehlende Benutzerauthentisierung:
  Weder in der Spezifikation noch in der Implementation ist eine Authentisierung eines Benutzers vorgesehen. Dies bedeutet im Falle des ZHW-Testnetzes, dass irgendjemand über dieses Testnetz in das ZHW-Schulnetz eindringen kann.

- Fehlende Paketauthentisierung:
  In der Spezifikation von "Cellular IP" ist vorgesehen, die Kontrollpakete zu authentisieren. Diese werden vom mobilen Teilnehmer über die Luftstrecke zur Basisstation und von dieser hinauf zu einem Gateway gesendet. Über eine Authentisierung kann verhindert werden, dass unliebsame Dritte mit falschen Kontrollpaketen den Netzbetrieb stören. Diese Paketauthentisierung wurde jedoch nicht in der Referenz-Implementation umgesetzt.

- Instabile Referenz-Implementation:
  Es hat sich herausgestellt, dass das Design sowie die Implementation der Software mit Namen "CIP v1.1 for Linux" mangelhaft ist, was höchstwahrscheinlich die instabile Arbeitsweise im Betrieb erklärt. Es müsste eigentlich eine Neuimplementation erfolgen.

Weil es sich nun bei "Cellular IP" um eine grossflächige Technologie handelt und damit eine grosse Benutzerbasis mit sich bringt, drängte sich hierzu eine Erstellung eines umfassenden Konzepts und damit eine Erweiterung der "Cellular IP"-Spezifikation geradezu auf. Das Konzept muss neben der Authentifizierung, der Autorisierung auch das Accounting enthalten ($\rightarrow$ AAA), um eine grosse Benutzerzahl ökonomisch sinnvoll verwalten zu können[1]. In diesem Zusammenhang müssen auch die unterschiedlichen Anwendungsfelder betrachtet werden, um eine universelle Lösung zu erarbeiten.

[1] Der Aspekt der Autorisierung spielt eine untergeordnete Rolle, weil in einem "Cellular IP"-Netz keine Ressourcen vergeben werden.

In dieser Projektarbeit wurde in einer ersten Phase ein Konzept erstellt, welches die erwähnten Eigenschaften behandelt. Das Ergebnis ist eine Spezifikation für eine weiterentwickelte Architektur von “Cellular IP”, die den Namen “Cellular IP next Generation” (→ Cellular IP nG) trägt. Das Konzept wird im Teil *Analyse* erarbeitet.

Da aus Zeitgründen nicht die gesamte erstellte Spezifikation umgesetzt werden kann, wurde in der Phase des Designs der zu realisierende Funktionsumfang abgegrenzt und untersucht, mit welchen Mitteln eine Implementation stattfinden kann. Dies ist deshalb von besonderer Bedeutung, weil eine direkte Erweiterung der Referenz-Implementation höchstwahrscheinlich in einer “Wegwerf-Implementation” endet, wenn aus dem obenerwähnten Punkt der Instabilität eine neue Fassung der Referenz-Implementation stattfindet. Die Diskussion zu diesen Punkten findet sich im Teil *Design*.

In der Phase der Realisierung wurde der definierte Funktionsumfang umgesetzt und schliesslich mit der Referenz-Implementation integriert. Die Beschreibungen der eingesetzten Technologien und Verfahren, die Modulbeschreibungen selbst sowie das Zusammenführen mit der Referenz-Implementation ist im Teil *Realisierung* widergegeben.

Der Teil *Projektverlauf* der Dokumentation zeigt auf, wie in diesem Projekt vorgegangen wurde sowie die Zeitplanung mit Soll-Ist Gegenüberstellung. In der Schlussbemerkung wird ein Resumée gezogen und ein Ausblick gegeben, in welche Richtungen sich “Cellular IP nG” weiterentwickeln könnte.

4 Grundlagen *Cellular IP*

In diesem Kapitel werden einige grundlegende Konzepte von “Cellular IP” erläutert, welche für spätere Betrachtungen notwendig sind. Für eine vollständige Beschreibung sei auf die vorhergehende Projektarbeit[1] sowie die Original-Literatur[2] verwiesen.

4.1 Terminologie

Base Station	Spezielle “Nodes”, welche zusätzlich zu kabelgebundenen Netzwerkkarten mit Wireless-LAN Karten ausgestattet sind.
Downlink	In der Hierarchie nach unten, in Richtung “Base Stations”, führender Link.
Handoff	Bezeichnung für den Wechsel der “Base Station”.
Mappings	Abbildung auf einem “Node”, welche die “Mobile Host”-Adresse einem Downlink zuweist, durch welches das letzte Pakete empfangen wurde.
Mobile Host	Transportables Gerät, welches mit einer Wireless-LAN Karte ausgerüstet ist.
Node	Dient als Konzentrator von Downlinks zu einem Uplink.
Page Update	Kontrollpaket, welches vom “Mobile Host” zum Gateway geschickt wird bei Inaktivität oder bei Handoffs.
Paging Cache	Auf bestimmten “Nodes” installiert; wird aktualisiert durch “Page Updates”.
Route Update	Falls “Mobile Host” aktiv, aber keine Daten sendet, so wird “Route Update” zum Gateway geschickt.
Routing Cache	Auf bestimmten “Nodes” installiert, wird aktualisiert durch “Route Updates”.
Uplink	In der Hierarchie nach oben, in Richtung Gateway, führender Link.

[1] http://www.strongsec.com/zhw/PA/PA1_Sna06_2001.pdf
[2] http://www.comet.columbia.edu/cellularip/pub/ccr99.pdf

4.2 Routing im “Cellular IP”-Netz

Um die Einfachheit und Skalierbarkeit in einem “Cellular IP”-Netz zu gewährleisten, kennt keiner der beteiligten “Nodes” den genauen Standpunkt eines anderen. Pakete, welche an einen “Mobile Host” adressiert sind, werden nach einem “Hop by Hop”-Verfahren zur “Base Station” geroutet, zu welcher der “Mobile Host” gerade eine Verbindung unterhält. Dabei kennt jeder beteiligte “Node” nur seinen unmittelbar nächsten Nachbarn, welcher er über seinen Downlink oder Uplink erreichen kann. Diese Routing-Informationen werden lokal in den “Nodes” gespeichert in sogenannten “Mappings”, welche eine “Mobile Host”-Adresse einem Downlink zuordnet. Diese “Mappings” werden erzeugt, wenn Daten vom “Mobile Host” in Richtung Gateway unterwegs sind, seien es Kontroll- oder Datenpakete. Um nun den Aufwand für das “Control Messaging” zu verringern, werden nach einem Handoff zu einer anderen “Base Station” die Mappings eines “Mobile Hosts” nicht gelöscht. Sie laufen zeitlich aus und werden danach entfernt. Daraus folgt, dass ein “Mobile Host” periodisch Paketen schicken muss damit, diese “Mappings” nicht gelöscht werden.

Es entsteht nun ein Problem, dass solange Daten vom Gateway in Richtung der alten “Base Station” gesendet werden, wie das “Mapping” des “Mobile Hosts” auf den traversierten “Nodes” durch den Timer noch nicht gelöscht wird.

Dieses Problem wird gelöst, indem zwei Arten von “Mappings” auf den “Nodes” eingeführt werden, welche in den sogenannten “Paging & Routing Caches” gespeichert werden. Der “Paging Cache” ist gedacht für inaktive “Mobile Hosts”, welche gerade keinen Paketfluss unterhalten, aber dennoch verbunden bleiben wollen. Der Timeout-Wert dieses Caches ist dem Häufigkeitsgrad von Handoffs angepasst. Die “Mobile Hosts” schicken periodisch vor Ablaufen dieses Timers ein “Page Update”-Paket hinauf zum Gateway. Die “Mappings” werden auf diese Weise aufgefrischt.

Der “Routing Cache” hingegen wird von aktiven “Mobile Hosts” unterhalten, welche gerade aktiv sind und damit einen Datentransfer durchführen oder auf Daten warten. Bei einem Datentransfer werden jegliche Pakete, welche vom Gateway oder vom “Mobile Host” stammen, benutzt, um das “Mapping” in den “Routing Caches” aufzufrischen. Falls ein “Mobile Host” aktiv ist, aber keine Daten zu senden hat, muss er periodisch “Route Update”-Pakete versenden, damit seine “Routing Cache Mappings” auf den “Nodes” nicht auslaufen.

Die Einträge in diesem “Routing Cache” laufen nach ca. 10 Sekunden aus, während die Einträge im “Paging Cache” mit rund 10 Minuten erhalten bleiben. Diese Werte können vom Operator frei auf den “Nodes” konfiguriert werden, um eine Performance-Optimierung zu erreichen.

4.3 Zusammenspiel Mobile IP & Cellular IP

“Cellular IP” wurde als Erweitung von “Mobile IP” entwickelt. Es wird deshalb nachfolgend das Zusammenspiel von “Cellular IP” und “Mobile IP” erläutert.

"Mobile IP"[3] erlaubt es einem "Mobile Host" auch von einem fremden Netz ("Foreign Network") ins Internet zu gelangen. Er benutzt hierzu seine vom "Home Network" zugewiesene IP-Adresse, um global erreichbar zu bleiben. Der "Mobile Host" meldet sich hierzu bei einem "Foreign Agent" an, welcher zum fremden Netz gehört. Dieser kontaktiert den "Home Agent" des "Mobile Hosts", welcher wie der Name schon andeutet im "Home Network" steht. Bei erfolgreicher Anmeldung wird ein Tunnel aufgebaut zwischen dem "Home Agent" und dem "Foreign Agent". Wenn der "Mobile Host" sich nun im Internet bewegt, so werden die Pakete, welche er erzeugt, über das fremde Netz ins Internet geroutet. Die Antwort-Pakete hingegen werden vom Internet ins "Home Network" geschickt. Der "Home Agent" sorgt nun dafür, dass diese Pakete über den Tunnel zum "Foreign Agent" gelangen, welcher diese dann zum "Mobile Host" weitergibt.

Dieses Verfahren ist sehr gut einsetzbar, wenn der "Mobile Host" über eine mehr oder minder statische Anbindung an das "Foreign Network" besitzt und somit die Trägheit der Anmelde- und Abmeldeprozedur nicht ins Gewicht fällt. Dies wird aber zu einem Problem, wenn eine Wireless-Lösung angestrebt wird. Hierbei sorgen hauptsächlich die Faktoren der Ortsveränderlichkeit des Teilnehmers und die hohe Bandbreite von Wireless LAN dafür, dass nicht einfach eine Antenne aufgestellt werden kann, um damit alle Teilnehmer eines Einzugsgebiets zu versorgen. Der Lösungsansatz hierfür ist wie beispielsweise bei GSM-Netzen die Aufteilung eines Gebiets in Zellen.

Hier kommt nun das baumartig strukturierte "Cellular IP" ins Spiel, welche einen Gateway als Wurzel und "Base Stations" als Blätter enthält. Der Gateway übernimmt die Funktion des "Foreign Agents", während die "Base Stations" die einzelnen Zellen bedienen. Wenn sich nun ein "Mobile IP"-Teilnehmer in einem "Cellular IP"-Netz anmeldet, so wird ein Tunnel nur ein einziges Mal vom "Home Agent" zum besagten Gateway aufgebaut, unabhängig davon, in welcher Zelle sich der Teilnehmer befindet oder bei Zellenwechsel befinden wird. Dadurch kann das statische "Mobile IP" um eine dynamische Seite erweitert werden.

4.4 Paketauthentisierung

In "Cellular IP" ist vorgesehen, dass wenn die "Base Stations" Pakete von "Mobile Hosts" empfangen, sie aus deren Absender-Adresse und dem "Network Key" einen "Secret Key" (PID) "on the fly" generieren, welcher das gemeinsame Geheimnis zwischen "Mobile Host" und "Cellular IP"-Netz darstellt. Mit Hilfe dieser PID erzeugen die "Base Stations" nun einen sogenannten "Message Authentication Code" (MAC) über den Timestamp und den Inhalt des empfangenen Pakets und vergleichen diesen MAC mit dem im Paket mitgelieferten. Falls beide übereinstimmen gilt das Paket als verifiziert.

[3] "IP Mobility Support" RFC 2002, Abschnitt 1.5

Teil II

Konzept-Analyse

5 Ausgangslage

In diesem Kapitel wird zuerst eine Übersicht zu “Cellular IP” gegeben, um dann einige Anwendungs-Szenarien zu diskutieren. Das Ziel dieser Szenarien ist das Ermitteln von Anforderungen, welche dann näher im Kapitel Problemfelder untersucht werden.

5.1 Übersicht

“Cellular IP” wurde wie bereits erwähnt als ergänzende Lösung zu “Mobile IP” entwickelt. Als grundlegende Schicht besitzt es die Wireless LAN Technologie gemäss IEEE 802.11. Damit stellt es einen Ansatz dar, wie ein Einzugsgebiet in Mikrozellen von 50 bis 100 Meter Durchmesser aufgeteilt werden kann, die mit einer Bandbreite von 11 MBit/s bedient werden. Wegen der kleinen Zellengrössen ergibt sich eine viel höhere Anzahl von “Base Stations” für die Abdeckung derselben Gebietgrösse verglichen mit einem GSM- oder UMTS-Netz. Diesen zusätzlichen Kosten, verursacht durch die erhöhte “Base Station”-Anzahl, wird versucht durch günstige Standard Computer Hardware entgegenzutreten, kombiniert mit geringeren Entwicklungskosten. Ein schlagendes Argument für ein “Cellular IP”-Netz trotz der erhöhten “Base Station”-Anzahl ist die enorme Bandbreite von 11 MBit/s über den Luftkanal. Zum Vergleich erreicht hier UMTS im voll ausgebauten Zustand gerademal 2 Mbit/s. In diesem Zusammenhang wird “Cellular IP” in die Kategorie der 3.5 beziehungsweise 4. Generationen-Netze gerechnet, während GSM ein 2. und UMTS ein 3. Generationen-Netz darstellen.

“Cellular IP” kann deshalb als eine zukunftsträchtige Technologie bezeichnet werden, die den Wunsch nach “Internet everywhere & everytime” erfüllen könnte. Damit sich jedoch ein solches grossflächiges Netz ökonomisch durchsetzen kann, muss es offen sein, um die vielfältigsten Anwendungsmöglichkeiten unterstützen zu können.

5.2 Anwendungszenarien

Eine bestechende Tatsache in einer “Cellular IP”-Umgebung ist, dass die Terminals, welche eine Anwendung erst möglich machen, bereits existieren. Dies ist nicht der Fall bei den Telefonie-Netzen: das GSM-Netz von vielen Providern wurde bereits auf GPRS umgerüstet, jedoch lassen die entsprechenden Terminals wie Handys und PC-Cards immer noch auf sich warten. Dieses Katz-und-Maus-Spiel wird sich bei der Einführung von UMTS höchstwahrscheinlich wiederholen.

Bei “Cellular IP” treten die Terminals in Gestalt von Notebooks oder “Personal Digital Assistants” (→ PDA) auf, welche ihre Drahtloskommunikationsfähigkeiten über eine Wireless LAN-PC-Card erhalten.

5.2.1 Notebooks

Die Preise von Notebooks sind mittlerweile an einem Punkt angekommen, bei dem sich die unterschiedlichsten Personen ein solches Gerät leisten können. Obwohl noch unhandlich im Umgang und schwer im Mitführen, können Benutzer solcher Geräte in Parks, Restaurants und auch in Zügen angetroffen werden. Nachfolgend werden stellvertretend zwei Anwendungsfälle mit Notebooks aufgezeigt:

Der mobile Student

Da gibt es beispielsweise den Studenten, welcher morgens seine E-Mails abfragt, danach in den Zug sitzt, um zur Schule zu fahren. Im Zug ruft er die aktuellsten Nachrichten aus dem Internet ab und sucht sich die neuesten Songs in MP3-Form zusammen und lädt sich diese herunter. Am Zielort angekommen, besucht er Vorlesungen und ruft hierzu im Hörsaal “on the fly” die entsprechenden Manuskripte der Dozenten ab.

“Cellular IP” tritt in diesem Anwendungsfall in drei Facetten auf:

1. der Student erhält einen Zugang in das “Cellular IP”-Stadnetz, welches bespielsweise durch “Base Stations” an Laternenpfählen erschlossen wurde und Zellendurchmesser von 50-100m besitzen.

2. Im Zug gelangt er über das “Cellular IP”-Zugnetz ins Internet. Hierbei befinden sich die “Base Stations” entweder im Zug mit ähnlichen Zellendurchmessern oder sie sind an den Leitungsmasten befestigt und bedienen wegen der hohen Geschwindigkeit ein grösseres Zellengebiet von 300 bis 500m Durchmesser. Im ersten Fall muss der Uplink der “Base Station” über einen gemeinsamen Kanal zu einem “Node” geschickt werden. Inwiefern sich hierzu die Oberleitung eignet bleibt abuuklären.

3. Auf dem Schulareal schliesslich, tritt “Cellular IP” als Campusnetz auf.

Anforderungen:

- Da diese drei Netztypen eigene “Adminstrative Domains” bilden, werden sie von unterschiedlichen Gateways und unterschiedlichen Providern bedient jedoch mit demselben Kunden. Es muss eine Provider übergreifende Kommunikation stattfinden.

- Der Download welcher im Zug gestartet wurde, muss auch bei einem Netzwechsel weitergeführt werden können.

- Der Student könnte als Abonnent oder Prepaid-Karten-Benutzer eines Providers auftreten → Die Abbrechnung mit anderen Providern erfolgt durch diesen Provider selbst.
- Bei beiden Accountingvarianten muss nach Datenvolumen oder Benutzungszeit abgerechnet werden können.

Der wartende Geschäftsmann

Ein weiterer Anwendungsfall kann auf dem Flughafen stattfinden. Ein Geschäftsmann muss auf einen Anschlussflug warten. In der Zwischenzeit nimmt er über das Internet Kontakt mit seiner Firma auf und fragt bespielsweise über eine SAP-Anwendung die neuesten Kennzahlen ab, welche er für seine Entscheidungsfindungen benötigt. Dadurch, dass er eine Spezialanwendung benötigt, welche auf seinem Notebook für ihn eingerichtet wurde, kann er diesen Datenaustausch nicht in einem Internet Café durchführen.

Anforderungen:

- Der Geschäftsmann will unkompliziert ins Internet gelangen. Er kauft sich hierzu eine Prepaid-Karte, mit welcher er beispielsweise für eine Stunde einen Internetzugang erhält.
- Der Geschäftsmann setzt eventuell “Mobile IP” ein, so dass der Gateway als “Foreign Agent” fungieren muss.

5.2.2 “Personal Digital Assistants”

Die PDAs etablieren sich mehr und mehr zu einem ständigen Begleiter. Dies belegen die zunehmenden Verkaufszahlen von Palm- und PocketPC-Geräten. Im Moment werden diese Geräte noch nicht serienmässig mit Drahtloskommunikationsfähigkeiten ausgerüstet, jedoch existieren bereits heute beispielsweise bei Compaq IPAQ PDAs PC-Card-Adapter, welche es erlauben Wireless LAN PC-Cards zu betreiben. Nachfolgend werden wieder zwei examplarische Anwendungen vorgestellt:

Der Risiko-Patient

Es gibt sehr viele Risiko-Patienten bei denen eine vierundzwanzigstündige Überwachung der Vitalfunktionen wünschenswert wären, ohne jedoch die Lebensgewohnheiten des Betroffenen dadurch zu beschneiden. PDAs erlauben hierzu das Sammeln und regelmässige senden von Körperdaten zu einem Arzt. In kritischen Fällen kann der PDA sogar einen Notarzt herbeirufen.

Anforderungen:

- Das Kommunikationsnetz muss eine ständige Verbindung gewährleisten.
- Da die Verbindungdauer zeitunabhängig ist, muss eine Abbrechung über das transferierte Datenvolumen erfolgen.

Die Killerapplikation

Der Vorteil der PDAs ist, dass sie allzeit ihrem Benutzer zur Verfügung stehen. Der Kundennutzen könnte enorm gesteigert werden, wenn diese elektronischen Begleiter als Kommunikationszentrum des Benutzers auftreten. Falls neben den erwähnten Anwendungen es gelingt, auch ein brauchbares VoIP zu realisieren, so hätte der Benutzer ein Universalgerät in der Hand. Denn nun wäre auch normale oder Bild-Telefonie möglich.

Anforderung:

- Das Netz darf keine zu grossen Delays und grössere Jitter erlauben.
- Da ein Sprachkanal ein relativ hohes Datenvolumen erzeugt, verglichen mit normalen Surfaktivitäten, sollte dieser Paketstrom nach Zeit und nicht nach Datenvolumen abgerechnet werden.

Aus diesen Betrachtungen und den Fähigkeiten von “Cellular IP” als Netztechnologie lassen sich einige Problemfelder identifizieren. Um nun die angestrebte Offenheit zu erreichen, werden deshalb im nächsten Kapitel einige Probleme des konventionellen “Cellular IP” diskutiert und mögliche Lösungen in Form von Erweiterungen aufgezeigt.

6 Problemfelder

Dieses Kapitel beschäftigt sich mit der Umschreibung der Probleme, welche sich bei der Umsetzung der Anwendungsfälle aus dem vorhergegangen Kapitel bei der Verwendung von “Cellular IP” ergeben. Es werden hierzu Lösungsansätze in Gestalt von Erweiterungen des “Cellular IP” diskutiert. Das Ziel dieser Bemühungen ist das Zusammenfassen aller Erweiterungen zu einer neuen Spezifikation und damit zu einem neuen Konzept, welches im nächsten Kapitel vorgestellt wird.

6.1 Authentisierung

6.1.1 Feststellung der Identität

Hierbei stellt sich das Problem, wie der “Mobile Host” seine Identität gegenüber dem Gateway beweisen kann.

Lösungsansätze:

- “Registration Number” mit Passwort:

  Der “Mobile Host” baut mit Diffie-Hellmann zuerst eine sichere Verbindung zum Gateway auf. Danach sendet er eine “Registration Number” mit Passwort zum Gateway. Dieser verifiziert die Daten beispielsweise bei einem RADIUS-Server.

  Der Vorteil dieser Lösung ist, dass die Prepaid-Karten als Rubbelkarten erstellt werden können. Auch kann auf diese Weise das Abonnentenprinzip einfach durchgeführt werden.
  Der Vorteil ist aber gleichzeitig ein Nachteil, denn die gesamten Benutzerdaten, sowie Accountinginformationen müssen auf einem RADIUS-Server abgelegt sein. Gerade bei tausendenen von produzierten, aber vielleicht noch nicht im aktiven Einsatz befindlichen Prepaid-Karten nimmt der zentrale Verwaltungsaufwand enorm zu.

- “Mobile Host”-Zertifikate:

  Der “Mobile Host” führt mit dem Zertifikat beim Gateway in der Registrations-Phase einen “Registration Request” aus. Der Gateway überprüft die Echtheit des Zertifikats und schaut bei der “Provider CA” nach, ob das Zertifikat nicht zurückgezogen wurde (“Revocation List”). Die Entscheidung, ob ein Zertifikat zurückgezogen werden muss, trifft der Gateway im Vorfeld anhand der Autorisierungs- und Accountinginformationen.

Das Zertifikats-Konzept ist nicht auf Prepaid-Karten beschränkt, sondern es kann auch parallel dazu im "Abonnentenmodus" eingesetzt werden. Hierzu besteht immer noch die Möglichkeit, einen externen Account-Server zu bemühen, dem der Gateway periodisch die gesammelten Kenndaten abliefert. Im Unterschied zu einer reinen Account-Server Strategie werden hier keine unbenutzten Accounts eingerichtet.

Gewählte Lösung:

Es wird die Zertifikatslösung verfolgt.

6.1.2 Feste "Mobile Host" IP-Adressen

Dadurch, dass das "Cellular IP"-Netz nur als Zugangsmedium in ein Providernetz dient, erwartet ein "Cellular IP"-Netz von den "Mobile Hosts", dass diese eine gültige IP-Adresse mitbringen. Wie wir gesehen haben, entspricht dies im Zusammenspiel mit "Mobile IP" der "Home Network"-Adresse.

Das Problem entsteht nun, wenn ein Festnetz-ISP seinen Internetzugang auch über "Cellular IP" anbieten will. Er muss nun jedem mobilen Kunden eine feste IP-Adresse zuweisen. Diese Situation verschlimmert sich, wenn der Provider Prepaid-Zugänge anbietet. Hier müsste jede Prepaid-Karte eine eindeutige IP-Adresse beinhalten, damit auch solche "Mobile Hosts" im "Cellular IP" identifiziert werden können. Besonders der letztere Fall führt zu einer Verschwendung von IP-Adressen: es werden tausende solcher Prepaid-Karten zum Verkauf angeboten, effektiv benutzt wird aber nur ein kleiner Prozentsatz.

Lösungsansätze:

- interner DHCP-Dienst:

  Der Gateway wird gleichzeitig als DHCP-Server betrieben, welcher die Adressvergabe steuert, während die "Nodes" als "DHCP Relay Agents" fungieren.

  Der Vorteil dieser Lösung ist, dass wie beim Festnetz Internetzugang keine statische Zuordnung der IP-Adressen auf "Mobile Hosts" stattfinden muss. Auch kann auf diese Weise der Gateway einen für sein Netz zugewiesenen Adressbereich selbständig verwalten.

  Dieser Ansatz bedingt, dass ein funktionierendes Subnetz-Routing vorhanden ist. Dies ist bei der gegenwärtigen Implementation beispielsweise nicht der Fall.

  Ein anderes schwerwiegendes Problem ist, dass theoretisch vier Pakete[1] ausgetauscht werden, welche unauthentisiert das "Cellular IP"-Netz durchqueren. Die Client-Pakete werden zudem noch über einen Netzbroadcast (255.255.255.255) gesendet, das heisst jeder "Node", welcher in der Verbindungslinie von der "Base

[1] Das ist ein DHCPDISCOVER (Broadcast) vom Client, eine DHCPOFFER vom Server danach ein DHCPREQUEST (Broadcast) vom Client und abschliessend ein DHCPACK vom Server.

Station" bis zum Gateway liegt, wird diesen Broadcast auf allen "Downlink Devices" erwidern. Dies setzt sich bei diesen sekundären "Nodes" fort, sodass das gesamte Netz überflutet wird.

- externer DHCP-Dienst:

  Dieser Ansatz unterscheidet sich vom ersten dahingehend, dass ein externer DHCP-Server hinter dem Gateway im Netz des Providers benutzt wird, welcher beispielsweise auch Festnetz-Zugänge bedient. Der Gateway wird somit selbst zu einem "DHCP Relay Agent".

  Falls die zentrale Verwaltung von IP-Adressen von einem Provider als notwendig erachtet wird, so kann dies bei diesem Ansatz als Vorteil gewertet werden.

  Im Übrigen gelten dieselben Nachteile wie beim ersten Ansatz.

- IP-Adressenvergabe durch Gateway:

  Hierbei verwaltet der Gateway einen ihm zugewiesenen Adressbereich selbständig. Nach Überprüfung des Zertifikats schaut der Gateway in einer Tabelle ("Mobile Host Address Table") nach, ob für die "Registration Number" des "Mobile Hosts" eine feste IP-Adresse zugewiesen ist[2]. Falls nicht, so wählt er selbständig aus dem ihm zugewiesenen Bereich eine IP-Adresse aus. Zusammen mit DNS IP-Adressen und Authentifikationsinformationen werden diese Daten zum "Mobile Host" geschickt, welcher daraufhin sein "Wireless Device" auf diese IP-Adresse setzt. In diesem Zusammenhang werden auch die DNS IP-Adressen gesetzt.

  Dadurch, dass die Adresse bereits in der "Registration Response" vorliegt, kann bereits ab diesem Zeitpunkt die Wireless-Strecke gesichert werden. Es wird somit nur noch das "Registration Request"-Paket ungesichert übertragen (siehe auch "Überflutungsangriff").

Gewählte Lösung:

Es wird die letzte Lösung verwendet jedoch mit einer Erweiterung: im Zuge der Benutzer-Authentisierung wird als "Registration Number" die eindeutige Seriennummer des "Mobile Hosts"-Zertifikats verwendet.

6.2 Autorisierung

6.2.1 Entziehen der Zugangsberechtigung

Wenn ein "Mobile Host" nicht mehr autorisiert ist, das "Cellular IP"-Netz zu benutzen, weil beispielsweise aus den Accountinginformationen hervorgeht, dass sein Guthaben aufgebraucht ist, so muss eine Möglichkeit bestehen, den Benutzer einerseits bei einer Neuanmeldung abzulehnen, als auch den Zugang während des eingeloggten Zustands zu unterbrechen.

[2] Dies feste Zuweisung ist vorallem für Firmenkunden interessant, da nun die Firmenmitarbeiter immer unter derselben Adresse erreichbar werden.

Lösungsansätze:

- Verteilen der Autorisationsinformationen:

  Wenn der Gateway einem “Mobile Host” die Zugangsberechtigung entziehen will, so schickt er eine Meldung an alle “Base Stations”[3] mit der Mitteilung den Verkehr des entsprechenden “Mobile Host” zu sperren.

  Dadurch, dass der Gateway alle “Base Stations” über seine “Downlink Devices” erreichen kann, genügt eine einfache Broadcastmeldung.

  Der Nachteil entsteht nun dadurch, dass die “Base Station” diese Sperrung auf unbegrenzte Zeit aufrecht erhalten müssen. Somit wird früher oder später der Speicher aufgebraucht sein. Auch die umgekehrte Lösung, dass die “Base Station” alle berechtigten Teilnehmer enthalten, ist keine Option, denn hier wird der Speicher einfach von Anfang an voll ausgelastet. Bei beiden Lösungen muss die Persistenz der Daten auf den “Base Stations” gewährleistet sein.

- Unterbrechen des Netzverkehrs:

  Da der gesamte Paketstrom aller angeschlossener “Mobile Hosts” durch den Gateway fliesst, kann dieser selektiv die Pakete eines nicht mehr autorisierten Teilnehmers wegwerfen.

  Der Vorteil ist, dass wirklich nichts mehr in das Providernetz beziehungsweise in das Internet gelangen kann und somit der Nutzwert für den Teilnehmer gleich null beträgt.

  Andererseits gelangt der “Mobile Host” immer noch in das “Cellular IP”-Netzwerk selbst, weil die “Base Stations” dessen MAC als gültig erachten.

- Wechsel der PID $\rightarrow$ “Network Key”:

  Ein “Mobile Host” gelangt nicht mehr in das Netz wenn die “Base Station” den MAC als ungültig erachtet. Der MAC wird aus dem Timestamp, dem Paketinhalt und der PID gebildet. Ihren eigenen Timestamp darf die “Base Station” nicht verändern, da sonst ebenfalls die Pakete von autorisierten “Mobile Hosts” als ungültig erachtet werden. Der Paketinhalt wird vom “Mobile Host” vorgegeben. Die PID jedoch wurde bei der Authentisierung vom Gateway aus der IP-Adresse des “Mobile Hosts” und des “Network Keys” gebildet. Die IP-Adresse kann der “Mobile Host” selber verändern, so dass die “Base Station” dieses nicht als Entscheidungskriterium nehmen kann. Wenn sich jedoch der “Network Key” ändert, so wird auch die “Base Station” einen anderen PID berechnen und damit einen anderen MAC erhalten.

  Der Vorteil ist, dass sich alle “Base Stations” auf denselben “Network Key” stützen und somit instantan der “Mobile Host” auch bei Handoffs bei keiner anderen “Base Station” mehr Zutritt ins “Cellular IP”-Netz erlangt.

  Da nun aber auch die PIDs der autorisierten Teilnehmer ändern, werden Ihre Pakete von den “Base Stations” abgelehnt werden. Um dieses Dilemma zu lösen, könnten sich die “Mobile Hosts” erneut beim Gateway anmelden, um eine neue

[3] Der Gateway kann alle “Base Stations” über seine “Leaf Devices” erreichen.

PID zu verlangen. Wenn dies nun auch der nicht mehr autorisierte “Mobile Host” versucht, so wird der Gateway diesen erkennen und seine Anmeldung ablehnen.

Gewählte Lösung:

Es werden die Lösungen zwei und drei kombiniert angewendet. Um eine häufige Verteilung des “Network Keys” zu unterbinden, wird dieser in einem festen Intervall beispielsweise zur vollen Stunde gewechselt. Der Verkehr der nicht mehr autorisierten “Mobile Hosts” wird bis zu diesem Zeitpunkt des Schlüsselwechsels aktiv durch den Gateway blockiert. Um zu verhindern, dass alle “Mobile Hosts” bei einem “Network Key”-Wechsel gleichzeitig über den Gateway hereinfallen und somit eine Spitzenbelastung sowohl beim Gateway als auch im Netz selbst verursachen, teilen die “Base Stations” den “Mobile Hosts” den Zeitpunkt des Schlüsselwechsels mit und dass dann ihre PID ungültig werden wird. Die “Mobile Hosts” wählen deshalb einen zufälligen Zeitpunkt zwischen dem Erhalt und dem Eintreten des Schlüsselwechsel, um ihre Anmeldung durchzuführen. Damit wird ebenfalls erreicht, dass der Netzverkehr eines “Mobile Hosts” ungestört noch mit der alten PID betrieben werden kann bis die neu PID eintrifft.

6.3 Accounting

6.3.1 Prepaid

Wie sich bei der Benutzerauthentisierung schon herausgestellt hat, ensteht bei der zentralen Verwaltung von “Prepaid Accounting”-Informationen sehr viel Verwaltungsaufwand. Besonders im Falle des Accounting wäre es wünschenswert, wenn der Gateway dahingehend entlastet werden könnte, indem er völlig zustandslos arbeitet. Dies bedeutet, dass alle relevanten Informationen vom “Mobile Host” gestellt werden.

Wo bei der Benutzerauthentisierung ein einfaches Zertifikat genügt, um die Identität eines “Mobile Hosts” zu überprüfen, müssen beim Accounting nun auch sensitive Daten, wie beispielsweise das Restguthaben beim “Mobile Host” hinterlegt werden. Damit Manipulationen nicht Tür und Tor geöffnet werden, müssen die Daten selbst versteckt und die Operationen auf diesen Daten über wohlbekannte Funktionen ausgeführt werden. Da ein Jedermann die Zugangssoftware für “Cellular IP” selbst schreiben kann, bleibt nur noch die Implementierung dieser Funktionen in Hardware.

Auf der Gateway-Seite stellt sich das Problem wie dieser ohne Führen von Zustandsinformationen über einen “Mobile Host” erkennen kann, ob dieser beispielsweise bei einer Abfrage des Restbetrages, nicht immer denselben Restbetrag als Bestätigung zurück schickt. Auch muss dafür gesorgt werden, dass eine solche Bestätigung nicht einfach zu einem späteren Zeitpunkt dem Gateway auf Verlangen vorgespielt werden.

Lösungsansätze für Medien:

- Floppy-Disks:

Da praktisch jedes Notebook ein Floppy-Laufwerk besitzt, würde es sich vom Verbreitungsgrad her eignen. Jedoch kann diese vom Manipulationsaspekt her gesehen höchstens für die Speicherung der Zugangsinformationen nicht aber der Accounting-Informationen herangezogen werden.

- Magnetkarten:

  Hier herrscht das gleiche Problem wie bei den Disketten. Nur existiert hier noch der Vorteil, dass gewisse Daten unkopierbar abgespeichert werden können, nicht aber unlesbar.

- Cryptokarten:

  Diese sind für solche Problemstellungen konzipiert und entwickelt worden. Sie erlauben das generieren eines Schlüsselpaares schon auf der Karte, wobei garantiert wird, dass der "Private Key" die Karte nie verlässt. Zusammen mit geschützten Speicherbereichen, welche nur durch Funktionen der Karte manipuliert werden können, eignet sich diese Karten als Accounting-Träger.

  Der Nachteil der Cryptokarten hingegen ist der relativ hohe Preis, so dass diese nicht als Wegwerfkarten gebraucht werden können. Dies könnte mit einem Depotkonzept jedoch gelöst werden.

Gewählte Lösung:

Für das Prepaid-Konzept wird das Cryptokarten-Konzept benutzt. Es wird hierzu eine Funktion benötigt, welche einen Timestamp und die Abbuchungshöhe als Argument übernimmt. Der Auslesewert enthält den übergebenen Timestamp, die übergebene Abbuchungshöhe sowie den Kartenrestbetrag inklusive einer über alle diese Daten mit dem "Private Key" erstellten Signatur zurück. Der Gateway wird nun periodisch Abbuchungsmeldungen verschicken, auf welche er innert einer kurzen Frist eine Bestätigung in Form eines "Mobile Host"-Zertifikats mit zusätzlichem Inhalt in Form des signierten Auslesewertes erwartet. Falls dieses nicht eintrifft, so wird er den Datenstrom des "Mobile Hosts" sperren (die Zugangsberechtigung entziehen). Dadurch, dass der "Private Key" nicht zugänglich ist, kann garantiert werden, dass in der Bestätigung für den Gateway weder das Guthaben noch der Timestamp verändert werden kann. Der Timestamp dient hierbei dazu, dass nicht einfach eine Bestätigung für den Gateway zwischenkopiert und diesem später vorgespielt werden kann. Falls der Gateway anhand einer Bestätigung feststellt, dass das Guthaben aufgebraucht ist, kann er bei der "Provider CA" das Zertifikat des "Mobile Hosts" als ungültig erklären lassen.

Was die Abbrechung selbst betrifft, so muss immer im voraus abgebucht werden, da ein "Mobile Host" sich bis zur nächsten Abbuchung vielleicht schon nicht mehr im Netz befindet. Dies kommt auch der Zustandslosigkeit des Gateways entgegen, da ansonsten die letzten Abbuchungen pro "Mobile Host" zwischengespeichert werden müssten. Das Verfahren kann in einer höheren Protokollschicht implementiert werden.

6.3.2 Abonnement

Beim abonnentenorientierten Dienst wird ein konventioneller Accounting-Server eingesetzt werden, dessen Problemstellung nicht Teil von “Cellular IP” sind.

Es muss jedoch dafür gesorgt werden, dass der “Mobile Host” das Verfahren für das Accounting nach der Benutzerauthentisierung, vorgeben kann.

Gewählte Lösung:

Die Authentisierung erfolgt hier ebenfalls über das Zertifikatsprinzip, nur dass ein anderer Accounting-Typ übermittelt wird. Dadurch erkennt der Gateway, dass er für diesen “Mobile Host” einen Accounting-Server kontaktieren muss.

Durch die Mitteilung der Accounting-Art können auf dem Gateway die unterschiedlichsten Accounting-Möglichkeiten realisiert und gleichzeitig eingesetzt werden.

6.4 Betriebssicherheit

6.4.1 Adressengebundene Paketauthentisierung

Da bei der von “Cellular IP” vorgesehenen Paketauthentisierung von einer eindeutigen IP-Absenderadresse ausgegangen wird, stellt sich das Problem, dass der IP-Vergabemechanismus nicht dem Schutz der Paketauthentisierung unterliegen kann. Weiter bedeutet dies, dass die “Base Stations” einen anfänglichen Netzwerkverkehr ohne Überprüfung ins “Cellular IP”-Netz lassen müssen.

Lösungsansätze:

- IPSec:

  Dieses Protokoll wurde genau für den Zweck der Paketauthentisierung und gar Paketverschlüsselung entwickelt. Es offeriert zwei Betriebsarten mit Namen “Transport & Tunnel Mode”. Der Transport Mode ist für die Anbindung eines “Road Warriors” (mobile Teilnehmers) an sein Firmennetz gedacht, während der Tunnelmode den gesamten Verkehr zwischen zwei Netzen tunnelt. Für “Cellular IP” ist der “Transport Mode” interessant, da hier ebenfalls “Mobile Hosts” als mobile Endgeräte vorkommen.

  Der Vorteil hierbei ist, dass neben der Paketauthentisierung auch die Benutzerauthentisierung durchgeführt werden kann. Im oben geschilderten Fall wird eine “Transport Mode” Verbindung vom “Mobile Host” zum Gateway aufgebaut. Mit Hilfe von Diffie-Hellman wird ein sicherer Kanal gelegt worüber dann die Konfigurationsdaten wie IP-Adresse dem “Mobile Host” mitgeteilt werden könnten.

  Damit dieser Ansatz jedoch funktioniert, müssen die Anworten wegen der “two way”-Kommunikation während der Diffi-Hellmann und der IKE Phase in IPSec über alle “Downlink Devices” gebroadcastet werden, da beim Weg vom “Mobile Host” zum Gateway wegen der fehlenden (ungültigen) IP-Absenderadresse keine Route aufgebaut werden kann.

- Eigenes “Registration Request / Response”-Protokoll:

  Der “Mobile Host” sendet bei der Anmeldung ein Registration Request-Paket hinauf zum Gateway. Die “Base Station” erkennt den Protokolltyp und lässt ihn durch. Der Gateway überprüft die Identität und schickt entweder ein “Registration Response”- oder “Registration Denied”-Paket. Danach kehrt er in den Ausgangszustand zurück. Die Antwort wird von der “Base Station” verifiziert und zum “Mobile Host” übertragen. Falls dieser das Paket nicht erhält, so wird er nach einem Timeout wieder von vorne beginnen.

  Der Vorteil dieser Variante ist, dass nur zwei Pakete benötigt werden, ohne weitere “Retransmission”-Kontrolle. Auch sind “Denial of Service”-Attacken auf das “Registration Request”-Paket beschränkt. Um den Pfad für die Rückantwort zu legen, könnte eine Well-Known IP-Adresse beim “Mobile Host” eingestellt werden, anhand derer die “Nodes” erkennen, dass sie die “Registration Number” als Routing-Kriterium verwenden sollen und nicht die Absenderadresse. Diese Zuordnung kann in einem “Registration Cache” stattfinden.

Gewählte Lösung:

Es wird die letzte Lösung verfolgt, mit der Erweiterung, dass die “Nodes” einen “Registration Cache” erhalten, in welchem die Route eines “Registration Request”-Paket festgehalten wird, um die Anwort des Gateways hinunterzurouten. Das Kriterium für einen Eintrag mit einem <“Registration Number”, “Downlink Device”> Tuppel ist das Vorkommen einer “Universal Mobile Host Address” ($\rightarrow$ UMH) als Absenderadresse. Der Lookup wird beim Auftreten dieser UMH-Adresse als Destinationsadresse benutzt.

6.4.2 Timestamp als Schutz vor “Replay Attacks”

In “Cellular IP” ist vorgesehen, dass in der Paketauthentisierung ein Timestamp statt einer Sequenznummer mitgeschickt wird. Dies hat den Vorteil, dass ein Dritter nicht einfach den Verkehr eines “Mobile Hosts” aufzeichnen kann, um diesen dann einer “Base Station” vorzuspielen. Dies wurde deshalb über eine Zeitmarke gelöst, weil bei einem Handoff eines “Mobile Hosts” keine Kommunikation unter den “Base Stations” stattfindet und somit nicht die aktuelle Sequenznummer ausgetauscht werden könnte.

Das Problem hierbei entsteht nun, dass eine gemeinsame Zeitbasis für die “Mobile Hosts” und alle “Base Stations” benutzt werden muss. Zusätzlich kommt hinzu, dass die Zeiteinstellungen des “Mobile Hosts” nicht verändert werden dürfen, da dieser vielleicht aus einer anderen Zeitzone stammt.

Gewählte Lösung:

Die Verteilung der Zeitbasis auf die “Mobile Hosts” erfolgt durch Übertragung eines Korrekturwertes von der “Base Station” zum “Mobile Host”, wenn die “Registration Response” vom Gateway geschickt wird. Indem der Paketstrom zum “Mobile Host” ebenfalls authentisiert wird, kann der Timestamp des authentisierten “Registration Response”-Pakets auf dem “Mobile Host” als Referenz-Timestamp verwendet werden. Um das “Ti-

me Shifting" der Zeitgeber zwischen dem "Mobile Host" und der "Base Station" auszugleichen, wird der "Mobile Hosts" die Timestamps der empfangenen authentisierten Pakete überprüfen. Wenn für eine gewisse Zeit keine Daten geschickt werden, so wird der "Mobile Host" eine Art authentisiertes Ping-Paket zur "Base Station" schicken, welche eine authentisierte Antwort mit dem aktuellen Timestamp schicken wird.

6.4.3 Verteilung des "Network Keys"

Um die Integrität des PID als "Secret Key" zwischen "Mobile Host" und "Cellular IP"-Netz ("Base Stations") zu wahren, muss in einem bestimmten Intervall, der "Network Key" gewechselt und "gleichzeitig" an alle "Base Stations" verteilt werden. Da nun auf der "Mobile Host"-Seite die PIDs ungültig werden, müssen die "Mobile Hosts" sich erneut beim Gateway melden.

Hierbei muss ein potentielles Problem beachtet werden, wenn eine "Base Station" A zu spät einen neuen Schlüssel erhält. Ein "Mobile Host", der sich bereits neu registriert hat und mit dem neuen PID arbeitet, wechselt nun zu dieser Station A. Diese wird die Pakete ablehnen und den "Mobile Host" zu einem erneuten "Registration Request" zwingen. Im schlimmsten Fall könnte der "Mobile Host" seine maximale Anzahl hintereinander stattfindenden "Registration Requests" überschreiten und damit die Verbindung ins Netz verlieren.

Lösungsansätze:

- Alle "Base Stations" besitzen einen gemeinsamen "Secret Key", welcher dem Gateway bekannt ist. Bei einem "Network Key"-Wechsel wird dieser den neuen "Network Key" mit diesem Schlüssel encrypten und broadcasten.

  Der Vorteil ist, dass der Gateway nur einen Schlüssel braucht. Es ist ein symmetrisches (schnelles) Verfahren. Zusätzlich braucht er die "Base Stations" nicht zu kennen.

  Um das ganze System jedoch nicht zu kompromittieren müsste dieser "Secret Key" selbst regelmässig gewechselt werden.

- Alle "Base Stations" besitzen einen "Private Key" und der Gateway kennt die "Public Keys" derselben. Er verschlüsselt den "Network Key" einzeln für jede "Base Stations" und schickt einen Broadcast.

  Der Vorteil ist, dass die Sicherheit nicht von einem "Secret Key" abhängt.

  Jedoch ergibt sich hier ein Aufwand für den Gateway, da dieser jede "Base Stations" verwalten muss.

- Der Gateway baut eine verschlüsselte Verbindung ("IPSec Transport Mode") zu jedem am "Downlink Device" angeschlossenen Knoten auf. Beide Knoten identifzieren sich gegenseitig mit Zertifikaten (über IKE). Danach übermittelt der Gateway den "Network Key" über diese gesicherte Verbindung. Diese Knoten werden

ebenfalls solche verschlüsselten Verbindungen zu angeschlossenen Knoten an ihren "Downlink Devices" aufbauen. Dies geht weiter bis die "Base Stations" nur noch als "Leafs" auftauchen.

Weil der Gateway die "Base Stations" nicht zu kennen braucht, wird er dadurch entlastet. Zudem muss er den Vorgang der Verschlüsselung nur für die Anzahl der an ihm angeschlossenen "Downlink Nodes" durchführen.

Der Nachteil ist, dass eigentlich korrekterweise jeder Knoten bei der "Provider CA" nachfragen sollte, ob das von seiner Gegenstelle erhaltene Zertifikat überhaupt gültig ist. Dies führt zu zusätzlichem Netzverkehr auf jeder Stufe der Hierarchie. Auch wird zwischen den Knoten jedesmal der ganze Diffie-Hellman-Key-Exchange durchgeführt, was zu einem gewissen Overhead führt.

Gewählte Lösung:

Die letzte Lösung wird mit dem nachfolgenden Ansatz ergänzt, um die beschriebenen Nachteile wettzumachen:

Am Anfang besitzen beide Partner einer IPSec-Verbindung nicht das Zertifikat des anderen. Deshalb werden beide bei der "Provider CA" nachfragen, ob das Zertifikat des jeweils anderen gültig ist. Bei einem OK wird ein Knoten dieses Partnerzertifikat zwischenspeichern und bei einem zukünftigen Verbindungsaufbau mit diesem Partner dessen gesendetes Zertifikat mit dem zwischengespeicherten vergleichen. Falls diese Zertifikate dieselben sind, so hat er die Gültigkeit selbst nachgewiesen[4]. Falls sie sich unterscheiden, so wird er erneut eine Anfrage bei der "Provider CA" starten, um das neue Zertifikat zu überprüfen. Falls dies OK ist, wird er das neue Partnerzertifikat zwischenspeichern und das alte löschen.

Der Nachteil dieser Lösung ist, dass der "Revocation"-Mechanismus ausgehebelt wird. Weil zwei Partner "Nodes" einander die ganze Zeit vertrauen, der eine aber mittlerweile durch eine Drittperson kompromittiert werden könnte, so hat der Provider keine Möglichkeit dem anderen "Node" über die "Revocation List" mitzuteilen, dass das Zertifikat nicht mehr gültig ist. Dies stellt jedoch ein prinzipielles Problem dar. Ein Knoten wird wahrscheinlich explizit neu aufgesetzt werden müssen, wenn dieser beispielsweise durch einen Hacker geknackt wurde oder der ganze Knoten selbst gestohlen wurde. In beiden Fällen wird der Operator ein neues Zertifikat verwenden. Dadurch dass der Partnerknoten wieder dessen Gültigkeit bei der "Provider CA" überprüfen wird, ist die Echtheit des Knotens garantiert.

6.4.4 Intrusiondetection

Hier stellt sich die Problematik wie eine "Base Station" oder ein "Node" erkennen kann, dass er Ziel einer Attacke ist. Diese Attacken können softwaremässig sein oder auch physisch am Gerät. Dieser Punkt ist deshalb von Bedeutung, weil für eine Erschliessung eines Stadtgebietes viele "Base Stations" "deployed" werden müssen, was eine zentrale Überwachung erschwert.

[4] Falls das Zertifikat jedoch ausgelaufen ist, so wird er die Verbindung ablehnen.

Es wird hier keine Lösung für diese Problematik erarbeitet.

6.4.5 Überflutungsangriff

Da die “Base Station” den “Registration Request” ungeachtet der Herkunft ins Netz hereinlassen muss, könnte der Gateway mit einer Flut von solchen Paketen in die Knie gezwungen werden.

Man kann von der Annahme ausgehen, dass im Einflussbereich einer “Base Station” nur ein bestimmter Prozentsatz von “Mobile Hosts” gleichzeitig versucht, sich beim Gateway zu registrieren. Daraus kann ein Intervall für eine “Base Station” bestimmt werden. “Registration Request”-Pakete, welche ausserhalb dieses Intervalls ankommen, werden von der “Base Station” “discarded”. Dadurch, dass die “Mobile Hosts” die “Registration Request”-Pakete in kürzeren Intervallen losschicken, sollte mindestens ein solches Paket unter eventuellen Mitkonkurrenten ins “Cellular IP”-Netz gelangen.

6.5 Infrastruktur

6.5.1 Störungen durch “Cellular IP”-Routing

In “Cellular IP” wird das normale Paket-Routing durch ein eigenes Routing ersetzt[5]. Hierzu werden Pakete welche im Uplink eines “Nodes” eintreffen nicht mehr gemäss Routing Tabelle geroutet, sondern es wird die Zieladresse des “Mobile Hosts” in einem Routing Cache das “Downlink Device” nachgeschlagen.

Während im Upstream die Pakete gemäss Defaultroute durch das “Cellular IP”-Netz hinaus geroutet werden können, funktioniert dies im Downstream nicht, weil die Adressen der “Mobile Hosts” in der Regel nicht einer Adresse im Subnetz des “Cellular IPs” entsprechen. Daraus folgt, dass das normale Subnetzrouting übergangen werden muss, weil sonst lauter “Destination unreachable”-Meldungen generiert werden. Hierbei besteht jedoch die Gefahr, dass Management Protokolle wie SNMP oder “Remote Logins” über *telnet* oder *ssh* nicht mehr verwendet werden können, um “Nodes” und “Base Stations” zu warten.

Lösungsansätze:

Generell ist zu sagen, dass mit der Zielsetzung einer koexistierenden Lösung beider Routingverfahren, eine Trennung der Adressräume von “Mobile Hosts” auf der einen Seite und der im “Cellular IP”-Netz verwendeten Geräte wie “Nodes” und “Base Stations” auf der anderen Seite stattfinden muss.

- Tunneln der “Mobile Host”-Paket im Upstream:

  Die Pakete der “Mobile Hosts” werden bei den “Base Stations” getunnelt und beim Gateway wieder enttunnelt.

[5] http://www.comet.columbia.edu/cellularip/overview.htm, Abschnitt 3.1

Der Vorteil ist, dass immer alle Pakete an den Gateway geschickt werden und somit niemals wegen einer Routingentscheidung des normalen Subnetz-Routings plötzlich in ein "Downlink Device" eines Knotens gesendet werden, wenn der "Mobile Host" mit einer Subnetz IP-Adresse konfiguriert wurde[6].

Ein Nachteil ist, dass ein Anfügen und Entfernen eines zusätzlichen IP-Headers für jedes Paket erfolgen muss, was eine gewisse Performanceverschlechterung mit sich bringt.

- Gatewayadresse als Defaultroute:

  Hier werden alle Pakete mit jedwelchen Destinationsadressen immer zum Gateway weitergeroutet. Gleichzeitig wird das normale Subnetz-Routing beibehalten.

  Der Vorteil ist, dass im Uplink-Pfad der Weg zum Gateway eindeutig vorbestimmt ist und somit ein Tunneling entfallen kann.

  Da nun hier die Destinationsadressen frei wählbar sind dürfen, sie nicht in das Subnetz von "Cellular IP" fallen. Um solche Pakete vorderhand ablehnen zu können, müssen die "Base Stations" solche Paket auf ihrem "Wireless Device" mit Hilfe der "Cellular IP"-Netzadresse identifizieren.

Gewählte Lösung:

Es wird die letztere Lösung verwendet, mit dem Zusatz, dass die "Wireless Devices" der "Base Stations" die Gateway-Adresse tragen. Damit muss auf den "Mobile Hosts" die Default-Adresse nur einmal eingetragen werden und das gesamte "Cellular IP"-Netz wie ein "collapsed Network" erscheint. Ein weiterer Vorteil dieser Erweiterung ist, dass der Operator sich nicht um die Verwaltung zusätzlicher IP-Adressen kümmern muss. Wenn nun "Nodes" "Downlink, Uplink & Wireless Devices" besitzen, so entsteht das Problem, dass die Datenpakete, welche auf dem "Downlink Device" eines "Nodes" eintreffen und gemäss Default-Route zum Gateway geschickt werden müssten, nun im "Node" selbst zum "Wireless Device" geroutet werden, weil dieses dieselbe IP-Adresse besitzt. Dies kann jedoch verhindert werden, indem generell solche Pakete abgefangen und selbst auf den Uplink geroutet werden[7].

6.5.2 Single Gateways

Beim normalen "Cellular IP" wird für ein bestimmtes Einzugsgebiet vom Einsatz eines Gateways ausgegangen. Es kann jedoch eine Aufteilung eines "Cellular IP"-Netzes in mehrere Netze sinvoll sein, wenn beispielsweise mehrere Ballungsgebiete erschlossen werden müssen. Auf diese Weise kann sowohl ein Loadbalancing als auch die Vermeidung von einem "Single Point of Failure" erzielt werden, da die "Mobile Hosts" physisch auf einer anderen Infrastruktur arbeiten.

[6] Dieser Fall kann ganz am Anfang der Anmeldeprozedur eintreten, wo der "Mobile Host" irgendeine Adresse besitzen kann, welche als "Mobile IP"-Adresse angesehen wird. Damit besteht aber hier die Gefahr, dass fälschlicherweise eine Adresse im Adressraum des "Cellular IP" eigenen Netzes konfiguriert wird.

[7] Dies kann mit "IP Tables" oder direkt in Netfilter gelöst werden.

Hierbei entstehen nun zwei Fälle, welche berücksichtigt werden müssen:

- Die Gateways kommen im selben Adressbereich zu liegen. Dies ist dann der Fall wenn ein Stadtnetz in mehrere Teilnetze aufgespalten wird.

- Die Gateways liegen in unterschiedlichen Adressbereichen. Dieser Fall tritt ein, wenn ein Provider ein "Cellular IP"-Netz in einer anderen Stadt errichten will, er selbst dort für den Backbone-Anschluss auf externen "Carrier" angewiesen ist.

Lösungsansätze:

- Gateways im selben Adressraum:

  Wenn die Gateways im selben Subnetz operieren, können sie dieselben IP-Adressen der "Mobile Hosts" gemeinsam verwalten[8]. Wenn nun ein "Mobile Host" einen Handoff in ein "Cellular IP"-Netz desselben Providers aber mit einem anderen Gateway durchführt, so muss der "Mobile Hosts" sich bei diesem neuen Gateway mit einem neuen "Registration Request" melden, um eine neue PID zu beziehen. Der ursprüngliche Gateway wird nun den Verkehr für diesen "Mobile Host" nicht mehr in sein "Cellular IP"-Netz hineinlassen.

- Gateways in verschiedenen Adressräumen:

  Bei einem Handoff eines "Mobile Hosts" in ein solches Netz muss derselbe sich beim neuen Gateway melden. Anhand des geänderten "Cellular IP Network Identifiers", welcher er über das "Beacon Signal" erhalten hat, wird der "Mobile Host" eine neue IP-Adresse vom Gateway verlangen. Hierbei ergeben sich zwei mögliche Varianten:

  1. Der neue Gateway teilt dem alten Gateway mit, dass er den gesamten Verkehr des "Mobile Hosts" zu ihm tunneln soll. Damit wird der alte Gateway zu einer Art "Home Agent" und der neue zum "Foreign Agent". Der Vorteil ist, dass bestehende Sessions oder Downloads des "Mobile Hosts" nicht abgebrochen werden müssen. Der Nachteil ist jedoch, dass der "Home Gateway" zusätzlich belastet wird.
  2. Der "Mobile Host" erhält einfach eine neue IP-Adresse. Der Vorteil ist, dass keine Interaktionen zwischen den Gateways notwendig ist. Jedoch werden bei dieser Lösung alle Sessions und Transfers abgebrochen.

Gewählte Lösung:

Beide Ansätze ergänzen sich und werden verfolgt.

[8] Diese "Cellular IP"-Netze könnten sich über denselben "Cellular IP Network Identifier" im "Beacon Signal" identifizieren.

6.5.3 Erkennung von “Cellular IP”-Netzen

Wenn mehrere “Cellular IP”-Netze von verschiedenen Providern übereinander liegen, so empfängt ein “Mobile Hosts” im “Ad Hoc”-Modus von mehreren “Base Stations” ein “Beacon Signal”. Nach “Cellular IP”-Spezifikation erfolgt eine Erkennung eines Netzes anhand des “Cellular IP Network Identifiers”. Damit alleine kann jedoch keine Aussage getroffen werden, welche Netze zu welchen Betreibern gehören.

Lösungsansätze:

- “Ad Hoc”-Modus:

  Das “Beacon Signal” wird um eine Operator-ID ergänzt, welche ebenfalls auf der Prepaid-Karte gespeichert wird. Die Applikation selbst trifft die Auswahl der “Base Stations” nach Kriterien des Operators und der Stärke des Signals.

- Infrastruktur-Modus:

  In diesem Modus wird das ESSID-Feld gemäss der 802.11-Spezifikation benutzt, um ein Netz eines Operators zu identifizieren. Die “Base Station” Wireless-Karte übernimmt selbständig das Erzeugen und Senden der “Beacon Signals”. Die “Mobile Host” Wireless-Karten wählen anhand der voreingestellten ESSID das entsprechende “Beacon Signal” aus um eine “Association” einzugehen.

  Der Nachteil dieser Lösung ist, dass kein direktes Roaming zwischen Providern stattfinden kann, weil die “Beacon Signals” transparent auf der Karte abgehandelt werden. Es ist jedoch möglich in einem Spy-Mode dennoch alle Aktivitäten zu protokollieren.

  Dadurch, dass ein Handoff zwischen Providern in der Regel weniger häufiger geschieht, als Handoffs zwischen Zellen desselben Providers, könnte eine Applikation eine Vorabselektion der “Beacon Signals” nach gültigen ESSIDs durchführen und bei einem notwendigen Handoff die Wireless-Karte mit dieser neuen ESSID konfigurieren.

Gewählte Lösung:

Konzeptionell wird die zweite Lösung verfolgt. Jedoch ist zu diesem Zeitpunkt der Betrieb der verwendeten Orinoco Wireless-Karte im Infrastruktur-Modus nicht möglich, weil dazu die entsprechende Firmware nicht offiziell zur Verfügung steht.

6.5.4 Verteilung der Zeit-Basis

Damit bei einem Handoff die neue “Base Station” die Pakete des “Mobile Hosts” wegen ungültigen Timestamps ablehnt, muss die Zeitbasis der “Base Stations” synchronisiert werden.

Dies kann mit einem “Time Distribution Protokoll” gelöst werden, wie zum Beispiel mit dem “Network Time Protocol” (->NTP)[9]. Dies setzt aber voraus, dass das Subnetz-

[9] http://www.eecis.udel.edu/~ntp/

Routing neben dem “Cellular IP”-Routing funktionsfähig ist. Andernfalls müsste hier eine eigene Implementation erfolgen.

7 Neues Konzept *Cellular IPnG*

7.1 Überblick

Um den Anforderungen, welche in der Ausgangslage angerissen wurden, gerecht zu werden sowie die Lösungen, welche in den Problemfeldern erarbeitet wurden umzusetzen, kann nicht umhin gekommen werden, "Cellular IP" in grösserem Massstab zu erweitern. Nachfolgend werden deshalb diverse Erweiterungen eingeführt, welche ein Tradeoff zwischen Kompatibilität zum konventionellen "Cellular IP" und der Verpflichtung zu den gestiegenen Anforderungen darstellt. Dieses Konzept hat die "Cellular IP"-Spezifikation zugrunde, jedoch wurde es bewusst nicht auf Verträglichkeit mit der Referenz-Implementation namens "CIP v1.1 for Linux" hin entwickelt wurden. Der Gund liegt in der instabilen Arbeitsweise dieser Applikation. Nach unserer Meinung müsste eine vollständige Neuimplementation der Software erfolgen. Aus der Performancesicht stellt sich ausserdem die Frage, ob nicht eine kernelbasierte Implementation sinnvoller wäre, statt eine Userspace-Applikation, wie in der Referenz-Implementation ausgeführt. Hier bietet sich der Netfilter-Mechanismus an, welcher im Kernelspace abläuft und erlaubt, Pakete vor dem Routing abzufangen, zu verändern und nach dem Routing-Mechanismus in den Paketfluss wiedereinzuspeisen.

In diesem Konzept wird aus Zeitgründen nur auf die Spezifizierung des Anmeldevorgangs sowie der Paketauthentisierung eingegangen.

7.2 Terminologie

Die Eigenschaften und Erweiterungen, welche in den Lösungen zutage gefördert wurden, können als Entitäten dieses Konzepts bezeichnet werden. Sie sind nachfolgend zusammengefasst:

"Accounting Charge Table" Diese Tabelle wird nur für das Nachführen der Charging-Aktivitäten im Prepaid-Modus auf dem Gateway benötigt. Es werden Trippel der Art <Mobile Host IP-Address, Time till Action, Action>. Als "Action" kommen "to be charged" und "to be verified" vor.

"Accounting Type" Die Art der Kontoführung wird im "Registration Request" dem Gateway mitgeteilt, so dass dort entsprechende Softwaremodule das Accounting übernehmen können. Jeder "Mobile Host" kann individuell mit einer Accountingart bedient werden. In diesem Konzept ist der Prepaid als Accounting-Typ ausführlich beschrieben.

"Beacon Signal" Dieses wird von den "Base Stations" im Infrastruktur-Modus transparent für höhere Layers verwendet, damit ein "Mobile Host" eine "Association" aufbauen kann. Es werden folgende Daten mitübertragen:

- "Cellular Network Identifier"
- IP-Adresse des Gateways
- "Paging Area"

Cryptocard Werden auf den "Mobile Hosts" eingesetzt und enthalten folgende Daten:

- Im X.509 Zertifikat:
  - "Operator ID"
  - Zertifikats-Seriennummer als "Registration Number"
  - "Public Key" des "Mobile Hosts"
  - Signatur der "Provider CA"
- Im geschützen Speicherbereich:
  - "Private Key" des "Mobile Hosts"
  - Accounting-Informationen bei Prepaid-Anwendungen
  - "Public Key" des Default-Operators
  - "Accounting Type"
- Im normalen Speicherbereich:
  - "Universal Mobile Host Address" (UMH-Adresse)[1]
- Die Karte enthält Funktionen für das Aufladen, Abfragen und Abbuchen vom Kartenguthaben.
  - Aufladen: Die Karte gibt eine Sequenzzahl zurück. Diese wird zum Operator geschickt, welche den Aufladungsbetrag und diese Sequenzzahl mit seinem "Private Key" signiert. Die Karte erhält diese Antwort und überprüft die Signatur sowie die Sequenzzahl. Falls beide OK wird Karte mit Betrag aufgeladen.
  - Abfragen: Parameter ist eine grosse Zahl (Randomzahl generiert vom Gateway). Rückgabewert ist Typ der Antwort (Abfrage), die übergebene Zahl, Flag ob Abfrage OK oder NotOK und Signatur über alle Daten erzeugt mit "Private Key".
  - Abbuchen: Parameter ist eine grosse Zahl (Randomzahl generiert vom Gateway). Rückgabewert ist Typ der Antwort (Abbuchung), die übergebene Zahl, Flag ob Abbuchung OK oder NotOK und Signatur über alle Daten erzeugt mit "Private Key".

Gateway-Adresse In jedem "Cellular IP"-Netz exisitiert nur ein Gateway. Dessen Adresse wird im "Beacon Signal" der "Base Stations" ausgestrahlt.
Die "Base Stations" verwenden ebenfalls diese Adresse um ihr "Wireless Device" zu bezeichnen. Damit wird einerseits der Administrationsaufwand verkleinert, andererseits erscheint das "Cellular IP"-Netz dann wie ein "collapsed"-Network.

[1] Diese Adresse ist im gesamten "Cellular IP"-Netz bekannt und kann vom Provider vorgegeben werden.

"Gateway Random Request" Wird gebraucht, um eine Zufallszahl vom Gateway anzufordern. Die Antwort erfolgt im entsprechenden Response-Packet.

"Gateway Random Response" Die Zufallszahl wird auf dem "Mobile Host" der "Cryptocard" für Abbuchungen oder Anfrage betreffend des Kartenrestguthabens der "Cryptocard" übergeben.

"Mobile Host Table" Diese Tabelle wird unter anderem benutzt, um die vergebenen IP-Adressen zu verwalten und wird auf dem Gateway eingesetzt. Sie enthält Einträge der Art <Mobile Host IP-Address, Registration Number, Mobile Host Public Key, Time till invalid, To Do>. Als "To Do" kommt " ip to be released" vor.

"Mobile Host"-Zertifikat Dieses dient der Benutzerauthentisierung im Anmeldevorgang.

"Network Key Change Phase" Diese Phase wird vom Gateway initiiert. Dazu schickt er den neuen "Network Key" mit dem Zeitpunkt des Wechsels an alle "Base Stations". Wenn dieser Zeitpunkt erreicht ist, starten dann die "Base Stations" die "PID Change Notification".

Diese Phase beginnt somit vor der Aktivierung des neuen "Network Keys" (z.B. 5 min) und geht eine gewisse Zeit über den Aktivierungszeitpunkt hinaus (z.B. 5 min).

"Operator ID" Im "Beacon Signal" wird das "SSID"-Feld explizit dafür benutzt, um einen Operator eines "Cellular IP"-Netzes zu identifizieren ("Operator ID").

"PID Change Notification" Die "Base Stations" schicken in einem bestimmten Intervall einen Broadcast in dem sie mitteilen, dass die PID auslaufen wird und bis wann eine neue Anmeldung durchgeführt werden muss. Der Start und der Stop dieser Meldung zeigt den Beginn beziehungsweise das Ende der "Network Key Change Phase" an.

"Provider CA" Der Gateway fragt bei der "Certification Authority" des Providers nach ob ein "Mobile Host"-Zertifikat noch gültig ist.

"Registration Cache" Die "Cellular IP"-Knoten müssen einen "Registration Cache" unterhalten, welcher nur für den "Registration Request" und "Registration Response" beziehungsweise "Registration Denied" benutzt wird. In diesem Cache werden Tuppel eingetragen der Art <Registration Number, Downlink Device>.

"Registration Number" Da am Anfang des Anmeldevorgangs unter Umständen noch keine "Mobile Host"-IP-Adresse existiert, wird das Routing der Antwortpakete mit dieser Nummer durchgeführt.

"Registration Response Buffer" Dieser Puffer ist für "Nodes" gedacht, welche mehrere "Base Stations" bedienen (beispielsweise mehrere in einer Kette). Er dient dazu, die "Registration Response"-Pakete so nahe wie möglich bei der "Base Station" zwischenzupuffern, welche das "Registration Request"-Paket weitergeleitet hat. Damit wird erreicht, dass wenn ein "Mobile Host" häufige Zellenwechsel vollzieht und dabei erneute "Registration Requests" absetzt, die Wahrscheinlichkeit enorm hoch wird, dass ein "Node", welcher die alte und die neue "Base Station" bedient, das "Registration Request"-Paket selbst sofort beantworten kann.

"Registration Request" Ermöglicht Registration des "Mobile Hosts" beim Gateway. Enthält Zertifikat des "Mobile Hosts" mit Zertifikats-Seriennummer als "Registration Number" sowie die Nachfrage nach dem verwendeten MAC-Algorithmus, der MAC-Grösse und optional weiterer Parameter.

Auf ein "Registration Request" kann eine "Registration Response" oder ein "Registration Denied" erfolgen. Um den Handoff im Zusammenhang mit dem "Registration Response Buffer" zu erleichtern, wird auch in der neuen Zelle ein "Registration Request" abgeschickt.

"Registration Response" Enthält positive Antwort des Gateways auf den "Registration Request" mit Angabe des zu verwendenen MAC-Algorithmus, der MAC-Grösse sowie weiterer Parameter.

"Registration Denied" Gateway schickt Ablehnung, falls in der Registrationsphase die Benutzerauthentisierung fehlschlägt.

"Registration Cancelled" Falls ein "Mobile Host" die Autorisation nach der erfolgreichen Anmeldung entzogen wird, weil beispielsweise sein Guthaben auf der Prepaid-Karte aufgebraucht ist, so wird vom Gateway diese Meldung geschickt.

"Registration Release" Wenn der "Mobile Host" sich ausloggt, so wird diese Meldung zum Gateway Gateway geschickt. Es erfolgt keine Antwort.

"state registered" Der "Mobile Host" ist in diesem Zustand, wenn er sich erfolgreich angemeldet hat.

"state unregistered" Der "Mobile Host" hat sich noch nicht angemeldet, hat sich abgemeldet oder wurde vom Netz ausgesperrt.

"Timestamp Update Request" Falls der "Mobile Host" keine Daten von einer "Base Station" empfängt, so schickt er diese Anfrage, um eine Antwort auszulösen, woraus er dann den neuen Korrekturwert bildet.

"Timestamp Update Response" Das Paket trägt den aktuellen Timestamp einer "Base Station".

"Verification Header" Dieser wird bei der Paketauthentisierung zwischen dem "Mobile Host" und der "Base Station" eingesetzt. Er enthält den "Message Authentication Code", Timestamp und Flags, welche für die "PID Change Notification" benötigt werden. Zurzeit werden folgende Flags eingesetzt:

- "Network Key Change Phase" (Base Station $\rightarrow$ Mobile Host)
- "Using New PID" (Mobile Host $\rightarrow$ Base Station)

UMH "Universal Mobile Host Address"; dies ist eine Default-Adresse im "Cellular IP nG"-Netz. "Mobile Hosts" benutzen diese um einerseits einen Retourpfad durch das Netz zu legen, als auch anzuzeigen, dass sie eine Adresse benötigen.

7.3 Ablaufbeschreibung

7.3.1 Anmeldevorgang

Ein Anmeldevorgang wird in folgenden Fällen ausgelöst unter der Bedingung, dass ein "Beacon Signal" zuvor empfangen wurde:

- Neustart des "Mobile Hosts"
- Restart der Software[2] auf dem "Mobile Host"
- Empfang desWechsel des "Network Keys"
- Handoff in neues "Cellular IP"-Netz desselben oder eines anderen Providers
- nach Verbindungsunterbruch
- Empfang einer "PID Change Notification"-Meldung

Normaler Ablauf

1. Die Wireless-Karte des "Mobile Hosts" empfängt "Beacon Signals" automatisch.

   - Im Normalbetrieb wählt die Wireless-Karte die "Base Station" desselben Operators selbst aus anhand der "Signal to Noise Ratio" (→ SNR). Die "Base Stations" desselben Operatornetz werden automatisch anhand der auf der Karte eingestellten ESSID identifiziert.
   - Falls die SNR nicht mehr den Erwartungen genügt schaltet die Software auf dem "Mobile Host" in den Spy-Mode, um auch "Beacon Signals" von "Base Stations" anderer "befreundeter" Operatornetze mit besserem SNR zu finden. Die Software wählt ein anderes Netz aus, indem sie die ESSID des entsprechenden "Beacon Signals" entnimmt und die Wireless-Karte konfiguriert. Nun läuft der obengenannte Punkt wieder ab.

   Die Software nimmt einen ARP-Cache Eintrag mit der Abbildung der Gateway-Adresse und der "Base Station"-MAC-Adresse vor[3] .

2. Falls keine feste Absender IP-Adresse[4] konfiguriert wurde, wird für die nachfolgenden Pakete die UMH als Sourceadresse und die Gateway als Destination-Adresse[5] eingetragen.

[2] Nachfolgend wird allgemein der Begriff Software benutzt, um die Implementation der beschriebenen Funktionalität zu bezeichnen. Konkret besteht dieses Programm aus Daemon-Diensten, Netfiltermodulen, GUI-Applikationen usw.

[3] Dies ist notwendig, da unter Umständen ein anderer "Mobile Host" die Gateway-Adresse fälschlich oder mutwillig konfiguriert hat und deshalb die MAC-Auflödung stören könnte.

[4] Feste IP-Adressen werden bei "Mobile IP" benutzt.

[5] Alle "Base Stations" besitzen dieselbe Adresse in der Gestalt der Gateway-Adresse.

3. Optional: Im Prepaid-Modus wird zu diesem Zeitpunkt statt eines “Registration Request” zuerst ein “Gateway Random Request” geschickt. Die “Gateway Random Response” wird benutzt, um den Betrag auf der “Cryptocard” abzurufen und eine signierte Antwort zu generieren. Falls die Karte ein NotOK zurückgibt so wird die Software vorderhand den Anmeldevorgang abbrechen und eine Fehlermeldung an den Benutzer ausgeben.

   Falls die Karte ein OK gibt, werden die Punkte 5 bis 7 und 9 bis 10 durchgeführt (dem “Registration Request”-Paket vorweggenommen).

4. Es wird ein “Registration Request” abgeschickt mit Angabe der Accounting-Methode. Im Falle von Prepaid wird die im Punkt 3 erzeugte signierte Antwort im “Registration Request”-Paket mitgeschickt.

5. Anhand der Gateway-Adresse und des ARP-Cache Eintrags auf dem “Mobile Host” wird die “Base Station” direkt bestimmt und das Paket zu dieser geschickt.

6. Das Paket wird von der “Base Station” anhand der Default-Route in Richtung Gateway geleitet.

7. Anhand der UMH-Adresse erkennen die dazwischenliegenden “Nodes”, dass es sich um ein Paket aus der Registrationphase[6] handelt.

   - Falls der “Registration Buffer” auf diesem “Node” aktiviert ist, wird mit der “Registration Number” und des Request-Typs zuerst nachgeschaut ob ein Response-Paket für das entsprechende Request-Paket existiert. Falls ja, so wird dieses zurückgeschickt und das ursprüngliche Request-Paket weggeworfen. Damit ist der Anmeldevorgang abgeschlossen. Falls nicht, so wird im nächsten Punkt weitergefahren.
   - Anhand der “Registration Number” wird ein Mapping <Registration Number, Downlink Device> im “Registration Cache” sowohl der “Base Stations”, “Nodes” und des Gateways aufgenommen. Die “Routing & Paging Caches” werden nicht verändert[7].

8. Beim Gateway angekommen werden folgende Aktionen durchgeführt:

   Generell gilt, dass wenn eine Anmeldung vom Gateway als ungültig erachtet wird, so wird die Anmeldung abgebrochen und eine “Registration Denied”-Antwort zum “Mobile Host” geschickt. Es obliegt dem “Mobile Host”, einen erneuten Versuch einzuleiten.

   a) Anfrage bei “Provider CA”, ob Zertifikat in der “Revocation List” enthalten ist. Falls ja, Meldung “Registration Denied” zum “Mobile Host” schicken.
   b) Accounting Typ bestimmen und Autorisation nachprüfen:
      - Prepaid-Modus: Der Gateway überprüft, ob in der signierten Antwort der “Cryptocard” die Random-Zahl der gerade noch vom Gateway verwendeten Random-Zahl entspricht. Falls nicht wird die Anmeldung abgebrochen. Weiter wird das Flag in der Antwort überprüft. Falls Flag NotOK

[6] Ein “Registration Request”- oder “Gateway Random Request”-Paket.
[7] Zu diesem Zeitpunkt ist noch nicht entschieden, ob der “Mobile Host” im Netz verbleiben darf oder nicht.

lautet, wird Anmeldung abgebrochen. Andernfalls wird Echtheit der Signatur in der "Cryptocard"-Antwort anhand des "Public Keys" im mitgelieferten "Mobile Host"-Zertifikat überprüft. Falls nicht authentisch wird die Anmeldung abgebrochen.

- Abonnenten-Modus: Es ist keine explizite Autorisierung notwendig. Falls Abonnement nicht mehr gültig sein sollte, so wird dies im Punkt a) anhand des zurückgezogenen "Mobile Host"-Zertifikats festgestellt.
- Sonstige Accounting-Methoden: Anhand des "Accounting Typs" im "Registration Request" kann zu diesem Zeitpunkt jedwelche Autorisierungsüberprüfungen vorgenommen werden.

c) Falls die Source-Adresse des Pakets eine UMH-Adresse ist, sucht der Gateway eine freie Adresse für den "Mobile Host"[8].

d) PID generieren aus "Network Key" und "Mobile Host"-Adresse.

e) PID mit "Public Key" des "Mobile Hosts" verschlüsseln.

f) Über dasselbe "Downlink Device" wie der "Registration Request" eingetroffen ist, wird nun eine "Registration Response" zum "Mobile Host" zurückgeschickt. Dieses enthält:

- UMH-Adresse als Destination-Adresse des IP-Pakets
- "Registration Number"
- Verschlüsselte PID
- neue IP-Adresse mit Subnetzmaske des Providernetzes, Domainname, DNS IP-Adressen

9. Dazwischenliegende "Nodes" empfangen das "Registration Response"-Paket vom "Uplink Device". Anhand der UMH-Adresse als Destination-Adresse, schlagen diese das "Downlink Device" im "Registration Cache" mit Hilfe der im "Registration Response" mitgelieferten "Registration Number" nach und schicken das Paket auf diesem weiter. Falls der "Registration Buffer" auf diesem "Node" aktiviert ist, wird das Response-Paket mit der "Registration Number" als Index für eine Zeit von beispielsweise fünf Sekunden zwischengepuffert.

10. Die "Base Station" versieht das Paket mit einem "Verification Header" und schickt es zum "Mobile Host".

11. Auf dem "Mobile Host" angekommen werden folgende Aktionen durchgeführt:

    a) Mit dem "Private Key" die PID entschlüsseln.

    b) Der im Paket mitgelieferte MAC wird mit einem auf dem "Mobile Host" erzeugten MAC verglichen. Dieser MAC wird mit der PID als Schlüssel und dem Inhalt aus Timestamp und "Packet Content" gebildet.

[8] Dieser Vorgang kann der Gateway an einen DHCP-Server delegieren, welcher ausserhalb des "Cellular IP"-Netzes steht. Anhand der "Registration-Number" kann der Gateway aber auch MHs immer dieselbe IP-Adresse zuweisen (beispielsweise für Firmenkunden). Dies ist eine Flexibilität für den Operator.

c) Falls der Vergleich negativ ausfällt wird ein neuer “Registration Request” gestartet.
Falls OK, wird die mitgelieferte IP-Adresse dem “Wireless Device” zugewiesen sowie die DNS Einstllungen, wie DNS IP-Adressen, Domainname, usw. eingetragen. Gleichzeitig wird ein Korrekturzeitwert aus dem eigenen und des im Paket mitgelieferten Timestamps gebildet.
→ Der “Mobile Host” befindet sich nun im Modus “registered”.

12. Falls die Absenderadresse keine UMH-Adresse war, wird der Gateway als “Foreign Agent” auftreten und “Advertisement”-Meldungen verschicken, so dass schlussendlich ein Tunnel vom “Home Agent” zum Gateway gelegt wird.

Handoff im selben Netz:

Es tritt ein Handoff im selben “Cellular IP”-Netz mit demselben Gateway ein (“Mobile Host” war “registered”):

Da die neue “Base Station” denselben “Network Key” benutzt, ist die PID immer noch gültig und somit bleibt der “Mobile Host” “registered”.

Handoff in ein anderes Netz (gleicher Provider):

Es tritt ein Handoff auf, in ein “Cellular IP”-Netz desselben Providers aber mit einem anderen Gateway (“Mobile Host” war “registered”):

1. Anhand der neuen Gateway-Adresse aus dem “Beacon Signal” führt der “Mobile Host” eine erneute Anmeldung durch, jedoch diesmal behält er die IP-Adresse bei. Die Anmeldung ist notwendig, weil in diesem “Cellular IP”-Netz ein anderer “Network Key” verwendet wird und die Pakete somit von der “Base Station” weggeworfen würden.

2. Der Gateway meint, dass es sich um einen “Mobile IP”-Host handelt und wird diesem “Advertisement”-Meldungen schicken.

3. Falls der “Mobile Host” explizit für “Mobile IP” konfiguriert wurde, so wird er dieses ganz normal behandeln.

4. Falls nicht tritt implizites “Mobile IP” in Kraft, das heisst der “Mobile Host” wird den alten Gateway als “Home Agent” definieren.

5. Der neue Gateway wird nun in jedem Fall als “Foreign Agent” auftreten, entweder zum echten “Home Agent” oder zum alten Gateway.

Wechsel des “Network Keys”:

1. Der “Network Key” hat gewechselt (“Mobile Host” war “registered”):
Die “Base Stations” erhalten den neuen Schlüssel, dieser tritt erst nach einer bestimmten Zeitspanne in Kraft (beispielsweise 10 Minuten). Während dieser Zeit holen sich die “Mobile Hosts” ohne Unterbruch des Datenstroms die neue PID.

a) “Die Base Stations” erhalten den neuen “Network Key”.

b) Sie senden periodisch beispielsweise alle 30 Sekunden eine “PID Change Notification” mit der noch verbleibenden Zeit (absolut in Form eines Timestamps) bis zum Auslaufen der alten PID zu den “Mobile Hosts”.

c) Die “Mobile Hosts” wählen einen zufälligen Zeitpunkt zwischen jetzt und der empfangenen Zeit minus 1 Minute[9].

d) Falls der Zeitpunkt gekommen ist, starten die “Mobile Hosts” einen “Registration Request”.

e) Der Rest läuft nach dem gleichen Schema wie ab 1.c).

f) Falls nun Daten vom “Mobile Host” verschickt werden, so wird die neue PID benutzt und dies im “Verification Header” mit einem Flag mitgeteilt. Daten von der “Base Station” werden mit der neuen PID authentifiziert.

g) Die “Mobile Hosts” setzen das Flag zurück, sobald sie den Zeitpunkt überschreiten, welchen sie im “PID Change Notification” als Timestamp erhalten haben.

7.3.2 Abmeldevorgang

1. Der “Mobile Host” will sich abmelden und schickt einmalig eine “Registration Release”-Meldung zum Gateway.

2. Auf dem Gateway wird das Statusfeld dieses “Mobile Hosts” in der “Mobile Host Table” auf “to be released” gesetzt. Falls die Meldung nicht eintrifft, so wird ein Timer auslaufen, weil keine Daten- oder “Page Update”-Pakete mehr eintreffen.

3. Nach dem nächsten “Network Key”-Wechsel werden alle solche Einträge entfernt.

7.3.3 Accounting

Prepaid

1. Nach erfolgreicher Anmeldung des “Mobile Host” mit dem Accounting-Typ “Prepaid” erstellt der Gateway einen Eintrag in der “Accounting Charge Table” mit der “Mobile Host”-Adresse, resetet den Timer und setzt den Status des Eintrags auf “to be charged”.

2. Falls der Zähler abgelaufen ist, so wird eine Verbindung[10] zum “Mobile Host” eröffnet.

[9] Damit auch der am spätesten registrierende “Mobile Host” keinen Unterbruch beim Schlüsselwechsel erfährt.

[10] Mit Verbindung ist hier ein TCP-ähnliche Verbindung gemeint, welche die Problematik der Retransmission und Paketverlusten beziehungsweise Paketverdoppelung regelt. Es könnte TCP selbst eingesetzt werden, jedoch ist der Overhead des Verbindungsauf- und abbaus zu gross verglichen mit den in diesem Fall übertragenen Nutzdaten.

3. Es wird eine “Charge Request”-Meldung zum “Mobile Host” geschickt mit einer Random-Zahl und dem abzubuchenden Betrag[11]. Gleichzeitig wird der Status des Eintrags in der obengenannten Tabelle auf “to be verified” gesetzt und der Timer geresetet.

4. Der “Mobile Host” führt auf der “Cryptocard” die Abbuchungsfunktion aus mit den aus der “Charge Request”-Meldung stammenden Random-Zahl sowie dem Betrag.

5. Der “Mobile Host” schickt die signierte Antwort der “Cryptocard” über die Verbindung zum Gateway.

6. Die Verbindung wird geschlossen.

7. Der Gateway überprüft mit dem “Public Key” des “Mobile Hosts” die Echtheit der signierten Antwort. Falls OK und das Flag in der Antwort ebenfalls OK lautet, dann wird der Timer geresetet und der Status wieder auf “to be charged” gesetzt.

   Falls die Antwort nicht authentisch ist oder der Timer in der Tabelle abgelaufen ist, so wird eine “Registration Cancelled”-Meldung zum “Mobile Host” geschickt und dessen Eintrag aus der “Accounting Charge Table” entfernt. Gleichzeitig wird in der “Mobile Host Table” der Eintrag des “Mobile Hosts” auf “to be released” gesetzt. Nach dem nächsten “Network Key”-Wechsel werden alle solche Einträge entfernt.

Abonnement

Nach erfolgreicher Anmeldung des “Mobile Host” mit dem Accounting-Typ “Abonnement” besteht eine Verbindung zu einem Account-Server. Der Gateway führt in einer anderen “Accounting Table” Buch beispielsweise über die eingeloggte Zeit den verursachten Datenstrom usw.

Im Pull-Betrieb wird der Account-Server alle Gateways abfragen und die Daten pro “Mobile Host” zusammentragen, während im Push-Betrieb alle Gateways die Daten periodisch zum Account-Server senden.

Hierbei wäre es denkbar, dass ein dritter Server die Daten im Pull- oder Push-Betrieb von den Gateways erhält und die Accounting-Daten, welche zum gleichen “Mobile Host” gehören “merged”. Diese aufbereiteten Daten werden danach in das Format des Accounting-Servers konvertiert und diesem im Push- oder Pull-Betrieb übermittelt.

7.3.4 Paketauthentisierung

Nach erfolgreicher Anmeldung ist der “Mobile Host” im Besitze einer IP-Adresse, der für Ihn präparierten PID und des Timstamp Korrekturwertes. Bis zum nächsten “Network Key”-Wechsel und der damit verbundenen “PID Change Notification” findet der Normalbetrieb statt.

[11] Der abzubuchende Betrag wurde bei der Anmeldung mit dem “Registration Request” mitgeschickt.

Hierzu wird vor jedes zu versendende Paket ein IP-Header sowie ein “Verification Header” mit Protokollnummer 150 angehängt. Nach dem “Verification Header” folgt das Originalpaket (inklusive eigenem IP-Header).

“Mobile Host” → “Base Station”:

1. Mit Hilfe der eigenen PID, des Timestamps und des Inhalts des abzuschickenden Pakets wird ein MAC gebildet. Dieser wird zusammen mit dem Timestamp in den “Verification Header” verpackt und vor das Originalpaket gehängt.

2. Auf der “Base Station” wird dieses Paket überprüft:

   a) Der Timestamp im “Verification Header” wird überprüft. Falls nicht OK wird das Paket weggeworfen.

   b) Aus der Absenderadresse im Paket und des “Network Keys” wird die PID des “Mobile Hosts” berechnet. Damit wird wie in Punkt 1 der MAC auch auf der “Base Station” gebildet.

   c) Der im Paket mitgelieferte MAC wird mit dem soeben erzeugten MAC verglichen. Falls sie übereinstimmen gilt das Paket als authentisch und wird zum Gateway hinaufgeroutet. Falls nicht wird es ohne Meldung an den “Mobile Host” weggeworfen.

“Base Station” → “Mobile Host”:

Es findet das obenbeschriebene Verfahren in umgekehrter Reihenfolge statt.

1. Die “Base Station” entscheidet anhand des “Routing Caches”, dass das Paket über ihr “Wireless Device” gesendet werden muss.

   a) Aus der Destinationsadresse im Paket und des “Network Keys” wird die PID des “Mobile Hosts” berechnet.

   b) Mit Hilfe dieser PID, des Timestamps und des Inhalts des abzuschickenden Pakets wird ein MAC gebildet. Dieser wird zusammen mit dem Timestamp in den “Verification Header” verpackt und vor das Originalpaket gehängt.

2. Auf dem “Mobile Host” wird dieses Paket zuerst überprüft:

   a) Der Timestamp im “Verification Header” wird überprüft. Falls nicht OK wird das Paket weggeworfen.

   b) Mit der eigenen PID, dem Timestamp und des Inhalts des empfangenen Pakets wird der MAC gebildet.

   c) Der im Paket mitgelieferte MAC wird mit dem soeben erzeugten MAC verglichen. Falls sie übereinstimmen gilt das Paket als authentisch und wird im IP-Stack des “Mobile Hosts” weitergegeben. Falls nicht wird es ohne Meldung an die “Base Station” weggeworfen.

7.3.5 PID Change Notification

Damit der “Network Key”-Wechsel ohne Unterbrüche bei den “Mobile Hosts” durchgeführt werden kann werden “PID Change Notification”-Meldungen versendet, sobald die “Network Key Change Phase” beginnt:

1. Die “Base Station” erhält vom Gateway den neuen “Network Key” sowie den Zeitpunkt des Wechsels.
2. Die “Base Station” beginnt zu einer bestimmten Zeit[12], welcher beispielsweise fünf Minuten vor dem Wechselzeitpunkt des “Network Keys” liegt, “PID Change Notifications” an jeden “Mobile Host” einzeln zu verschicken[13], mit der Angabe des absoluten Wechselzeitpunktes in Timestamp-Einheiten. Gleichzeitig werden nun bei allen Paketen, die die “Base Station” verlassen im “Verification Header” das Flag “Network Key Change Phase” gesetzt.
3. Der “Mobile Host” verifiziert das Paket wie im Abschnitt 7.3.4 beschrieben.
4. Der “Mobile Host” erkennt die Meldung als Aufforderung für eine Neuanmeldung. Er wird nun einen zufälligen Zeitpunkt Wählen zwischen der Zeitspanne ab jetzt bis zum Wechselzeitpunkt, welcher in der Meldung angeben ist.
5. Beim Eintreten dieses gewählten Zeitpunktes wird eine neue Anmeldung, wie in Abschnitt 7.3.1 beschrieben, durchgeführt, um die neue PID zu erhalten[14].
6. Der “Mobile Host” wird nun für ausgehende Pakete im “Verification Header” das Flag “Using New PID” setzen. Anhand dieses Flags erkennt die “Base Station”, dass das Paket mit der PID aus dem neuen “Network Key” überprüft werden soll.
7. Nachdem die “Network Key Change Phase” vorüber ist, setzt die “Base Stations” für alle ausgehenden Pakete das Flag “Network Key Change Phase” im “Verification Header” zurück.
8. Der “Mobile Host” erkennt dies und setzt seinerseits das Flag “Using New PID” zurück.

→ Beide Stationen befinden sich nun wieder im Ausgangszustand.

7.3.6 Zeit-Synchronisierung

Mobile Host ↔ Base Station

1. Variante

Die “Base Station” sendet im “Beacon Signal” ihren aktuellen Timestamp als Referenz-Zeit mit. Der “Mobile Host” kann diesen Timestamp benutzen, um einen Korrekturwert

[12] Dieser Zeit kann beispielsweise fünf Minuten vor dem Wechselzeitpunkt liegen.
[13] Es muss jeder Einzelne adressiert werden, damit diese das Paket verifizieren können.
[14] Weil der Gateway selbst den “Network Key Change” angestossen hat, darf er eintreffende “Registration Requests” mit der PID beantworten, welche aus dem neuen “Network Key” gebildet wurde.

zu bilden. weil die Zeitbasis auf beiden Stationen eine unterschiedliche Ausgangslage haben.

Falls der “Mobile Host” einen “Time Shift” der Zeitbasen feststellt so berechnet er erneut den Korrekturwert anhand des Beacon-Timestamps.

2. Variante

Da die Zeit-Basis auf den beiden Stationen unterschiedlich ist, wird auf dem “Mobile Host” zuerst ein Timestamp Korrekturwert berechnet aus der Differenz des eigenen Timestamps sowie des Timestamps aus dem “Verification Header” des “Registration Response”-Pakets.

Um den “Time Shift” der Zeitgeber selbst zu kompensieren, wird der “Mobile Host” bei überschreiten eines vordefinierten maximalen “Time Shifts” den obenerwähnten Korrekturwert erneut berechnen anhand des Timestamps aus dem “Verification Header” irgendeines von der “Base Station” empfangenen *verifizierten* Pakets.

Falls keine Daten empfangen werden, weil der “Mobile Host” inaktiv ist, wird periodisch ein “Timestamp Update Request” geschickt, um eine Antwort seitens der “Base Station” zu provozieren.

Base Stations ↔ Gateway

Damit der Timestamp-Mechanismus in der Paketauthentisierung funktioniert müssen alle “Base Stations” dieselbe Zeit-Basis besitzen, weil sonst bei einem Handoff die Diskrepanz des “Mobile Host” und der neuen “Base Station” zu gross sein könnte. Eine erneute Anmeldung wäre notwendig, was konzeptionell nicht akzeptabel ist.

Indem in diesem Konzept das Subnetz-Routing neben dem “Cellular IP”-Konzept koexistieren kann, wird die Verwendung eines “Time Distribution Protocols” ermöglicht.

Eine Möglichkeit wäre das UDP-basierende “Network Time Protocol”[15], welches in einem WAN-Netz eine maximale Abweichung von 10- 20 ms erreicht.

Leider kann dieses Protokoll nicht verwendet werden, um ebenfalls die Synchronisation zwischen den “Mobile Hosts” und der “Base Stations” zu lösen. Der Grund liegt hier in der dreistündigen Einschwingphase, nach welcher erst diese Synchronizität erreicht wird.

7.3.7 Routing

Der Mechanismus der “Routing & Page Caches” wird komplett vom “Cellular IP” Originalkonzept übernommen.

[15] http://www.eecis.udel.edu/~ntp/

7.3.8 Network Key Distribution

Die Lösung für die Verteilung des “Network Keys” wurde in der Problemfeldanalyse ausführlich diskutiert. Nachfolgend ist die für das neue Konzept von “Cellular IPnG” gewählte Lösung festgehalten.

Um die Integrität des PID als “Secret Key” zwischen “Mobile Host” und “Cellular IP”-Netz (“Base Stations”) zu wahren, muss in einem bestimmten Intervall der “Network Key” gewechselt und “gleichzeitig” an alle “Base Stations” verteilt werden. Da nun auf der “Mobile Host”-Seite die PIDs ungültig werden, müssen die “Mobile Hosts” sich erneut beim Gateway melden.

Am Anfang besitzen beide Partner einer IPSec-Verbindung nicht das Zertifikat des anderen. Deshalb werden beide bei der “Provider CA” nachfragen ob das Zertifikat des jeweils anderen gültig ist. Bei einem OK wird ein Knoten dieses Partnerzertifikat zwischenspeichern und bei einem zukünftigen Verbindungsaufbau mit diesem Partner dessen gesendetes Zertifikat mit dem zwischengespeicherten vergleichen. Falls diese Zertifikate dieselben sind, so hat er die Gültigkeit selbst nachgewiesen[16]. Falls sie sich unterscheiden, so wird er erneut eine Anfrage bei der “Provider CA” starten, um das neue Zertifikat zu überprüfen. Falls dies OK ist, wird er das neue Partnerzertifikat zwischenspeichern und das alte löschen.

Falls eine “Base Station” beispielsweise durch “Intrusion” kompromittiert wird muss nach Bekanntwerden dieses Zwischenfalls das Zertifikat derselben zurückgezogen werden. Es muss daraufhin ein neues RSA-Schlüsselpaar sowie ein neues Zertifikat erstellt und auf die “Base Station” installiert werden[17]. Dadurch, dass der Partnerknoten wieder dessen Gültigkeit bei der “Provider CA” überprüfen wird, ist die Echtheit des Knotens garantiert.

7.4 Probleme / Schwächen

Da die “Registration Request”- und “Gateway Random Request”-Pakete nicht der Paketauthentisierung unterliegen können, muss die “Base Station” diese zwangsweise durchlassen. Hier bietet sich aber eine Angriffsmöglichkeit für Überflutungsangriffe.

Eine mögliche Lösung ist, dass die “Mobile Hosts” subsequentiell mehrere Pakete schicken und ein “Leaky Bucket”-Verfahren auf der “Base Station” dafür sorgt, dass keine Überflutung des Netzes stattfindet.

[16] Falls das Zertifikat jedoch ausgelaufen ist, so wird die Verbindung ablehnen.

[17] Eine elegante Lösung kann erreicht werden, wenn auch bei den “Base Stations” “Cryptocards” eingesetzt werden, da dann die Installation sich auf das Auswechseln der Karte beschränkt.

Teil III

Software-Design

8 Allgemeines

8.1 Einschränkung Funktionsumfang

Da das Konzept zu umfangreich ist, um dies in der gegebenen Zeit komplett zu implementieren, muss das Software-Design selbst auf einen realisierbaren Umfang reduziert werden.

Als Hauptfunktionsgebiet wurde die Paketauthentisierung ausgewählt, da diese instantan die Sicherheitssituation der vorhandenen Referenz-Implementation verbessert, somit im Normalbetrieb einsetzbar ist. Zudem kann der Anmeldevorgang überbrückt werden, indem die PIDs vom Systemadministrator manuell anhand der “Mobile Host”-Adresse und des “Network Keys” generiert und auf den “Mobile Hosts” manuell konfiguriert werden. Der Normalbetrieb kann dann aber ohne Konfigurationseingriffe erfolgen.

Für die Zeitsynchronisation wurde die 2. Variante ausgewählt obschon aufwendiger. Die erste Variante setzt einen Betrieb der Wireless-Karten im Infrastruktur-Modus voraus. Zum Zeitpunkt dieser Projektarbeit war diese Betriebsart wegen fehlender Firmware nicht möglich. Zu diesem Umstand kommt hinzu, dass die Referenz-Implementation die Karten im “Ad Hoc”-Modus betreibt sowie ein eigenes nicht konformes Layer 2 “Beacon Signal” versendet, was eine saubere Integration erschweren würde.

9 Entwurf

9.1 Lösungsansätze

Während der Einarbeitungsphase wurden Feasability-Tests durchgeführt, aus welchen sich drei mögliche Realisierungsansätze für die Softwarelösung[1] ergeben. Es wurden hierbei die Möglichkeiten der Paketveränderung und Paketweiterleitung untersucht.

1. "Packet Capture" mit LIBPCAP:

   Diese Bibliothek erlaubt es Pakete direkt vom "Network Device" zu lesen und zu schreiben. Es können Filter gesetzt werden, um beim Empfang beispielsweise nach Protokoll zu filtern.

   Der Vorteil ist, dass die Applikation mit herkömmlichen Bibliotheken geschrieben werden kann.

   Der Nachteil ist, dass es diese Applikationen im Userspace laufen. Damit sind Kontextwechsel für jedes empfangene Packet notwendig, was zu Performance--Einbussen führt.

2. IP-Stack verändern mit Kernel-Patch:

   Der IP-Stack wird mit Funktionsaufrufen ergänzt. Diese Funktionen werden erst von ladbaren Modulen dann gesetzt. Die Module implementieren dann, das Verhalten für Paketüberprüfung & -manipulation.

   Der Vorteil ist, dass jedes Paket erfasst wird: kein Paket wird unverifiziert den IP-Stack hinaufgehen.

   Da dies faktisch Änderungen am Kernel bedeuten, besteht die Verpflichtung den Patch regelmässig zu überprüfen und zu aktualisieren.

3. Netfilter Framework:

   Hierbei handelt es sich um eine Ergänzung des normalen IP-Stacks, welcher die Idee aus dem 2. Punkt aufgreift. Es handelt sich jedoch um ein generisches Konzept welches auch "iptables" oder "NAT" als Grundlage benutzen. Die eigene Funktionalität kann an sogenannten "Hooks" angehängt werden, welche verschiedene Stadien der Paketbehandlung des IP-Stacks darstellen.

   Dieses Framework ist das Resultat einer Überarbeitung des Filtermechanismus der Kernel 2.2 auf Kernel 2.4. Es wird somit in künftigen Kernelversionen enthalten sein.

[1] Damit ist die zu realisierende Software gemeint.

Was die Evaluation der Umsetzungsmöglichkeiten selbst betrifft, so haben sich hier zwei Ansätze herauskristalliert:

1. Die Referenz-Implementation selbst wird erweitert:

   Dies hat den Vorteil, dass direkt diejenigen Stellen, wo die Kontrollpakete bearbeitet werden, diese authentisiert beziehungsweise verifiziert werden können.

   Wie bereits erwähnt wurde, besteht hierbei jedoch die Gefahr, dass die Erweiterungen bei einer Neuimplementation der gesamten Software hinfällig werden.

2. Es wird ein "Dummy Device" verwendet, um die Paketveränderungen vor der Referenz-Implementation zu verbergen:

   Wenn ein Paket vom Gateway eintrifft, so liest die Softwarelösung von der Wireless-Karte das Paket ein, verifiziert es und schreibt es in das "Dummy Device". Die Referenz-Implementation liest nun das Paket vom "Dummy Device". Ein zu sendendes Paket wird in das "Dummy Device" geschrieben, worauf die Softwarelösung diese liest, das Paket authentisiert und auf die Wireless-Karte schreibt.

   Der Vorteil ist, dass die Referenz-Implementation nicht verändert wird und somit als potentielle Fehlerquelle wegen des Eingriffs ausgeschaltet werden kann.

   Es entsteht jedoch ein gewisser Overhead, weil pro Paket ein zusätzliches Lesen und Schreiben stattfindet.

Gewählte Lösung

Aus den vorhergehenden Betrachtungen wird der Lösungsansatz mit dem "Dummy Device" verwendet.

Paketauthentisierung

Betreffend der technischen Umsetzung wird der Paketauthentisierungs-Mechanismus in den Kernelspace verlagert und mit "Netfilter" realisiert. Dies bringt einerseits einen Performance-Vorteil, da zusätzliche Kontextwechsel entfallen. Dadurch, dass die Paketauthentisierung an den letzten "Hook" gehängt wird in der Reihe der Verarbeitung für zu sendende Pakete, kann dieses Modul wiederverwendet werden.

Dies wird besonders in den folgenden beiden Fällen interessant:

- Das "Netfilter"-Konzept prädestiniert sich für Paketbehandlung und deshalb ist eine Neurealisierung der Referenz-Implementation mit diesem Verfahren sehr wahrscheinlich.
- Die Lösungsansätze des "Cellular IP nG" lassen sich ebenfalls modularisieren und "Netfilter"-basierend implementieren. Damit tritt die in dieser Projektarbeit realisierte Paketauthentisierung als effektiver Beitrag zum neuen Konzept auf.

Forwarding Dummy Device $\leftrightarrow$ Wireless Device

Ein Feasability-Test hat sich herausgestellt, dass wenn die Referenz-Implementation auf das "Dummy Device" schreibt, ein "Netfilter"-Modul dies nicht zu Gesicht bekommt. Daraus lässt sich schliessen, dass das Paket nicht den IP-Stack hinauf geschickt wird.

Jedoch hat ein anderer Feasability-Test ergeben, dass mit LIBPCAP diese von der "Referenz-Implementation stammenden Pakete gelesen werden können. Dieser "Forwarding"-Mechanismus zwischen den Devices wird deshalb mit einer LIBPCAP-Lösung durchgeführt.

9.2 Aufteilung in Software-Module

Für die Realisierung entstehen somit zwei Software-Module. Die umzusetzende Funktionalität ist nachfolgend zusammengefasst:

Software-Modul "Verification":

- Paketauthentisierung für alle Pakete (Kontroll- und Datenpakete)
- Zeitsynchronisierung zwischen "Base Station" und "Mobile Host"

Software-Modul "Forwarder":

- Bindeglied zwischen Referenz-Implementation und Software Modul "Verification"
- Umleiten aller Pakete von einem "Dummy Device" zur Wireless-Karte und umgekehrt

Teil IV

Realisierung

10 Grundlagen

10.1 Netfilter

Netfilter ist ein “Framework” für das Modifizieren von Paketen ausserhalb des normalen “Berkeley Socket Interfaces”. Es definiert fünf Hooks, die sich an bestimmten Positionen im Protokoll-Stack von Linux befinden:

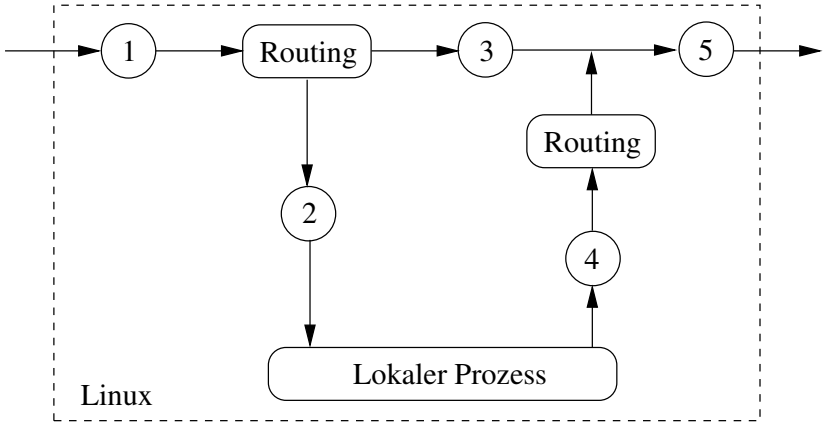


Abbildung 10.1: Netfilter Architektur

NF_IP_PRE_ROUTING (1)	Nachdem das Paket ins System eingetreten ist.
NF_IP_LOCAL_IN (2)	Bevor das Paket an einen lokalen Prozess übergeben wird.
NF_IP_FORWARD (3)	Wenn das Paket zu einer anderen Netzwerkkarte weitergegeben wird statt. zu einem lokalen Prozess.
NF_IP_LOCAL_OUT (4)	Nachdem ein lokaler Prozess das Paket gesendet hat.
NF_IP_POST_ROUTING (4)	Bevor das Paket das System verlässt.

An die verschiedenen Hooks lassen sich “Callback”-Funktionen binden, die für jedes den Hook passierende Paket aufgerufen werden. Falls mehrere Funktionen auf dem selben Hook sind, so wird anhand einer einstellbaren Priorität bestimmt, welche Funktion zuerst gestartet wird. Haben mehrere Funktionen die gleiche Priorität, sie wird diejenige zuerst gewählt, die zuerst registriert wurde. Um eine “Callback”-Funktion an einem Hook an- beziehungsweise abzumelden müssen die Funktionen “nf_register_hook” und “nf_unregister_hook” verwendet werden.

Durch den Rückgabewert der “Callback”-Funktion wird der weitere Weg eines Paketes bestimmt:

NF_ACCEPT	Das Paket wird normal weitergegeben.
NF_DROP	Das Paket wird weggeworfen.
NF_STOLEN	Das Paket ist von der “Callback”-Funktion übernommen worden, der Speicher wird vom System nicht freigegeben.
NF_QUEUE	Das Paket wird in eine Queue gelegt und für dazu registrierte “Callback”-Funktionen bereitgestellt.
NF_REPEAT	Das Paket wird nochmals am Anfang des Hooks eingespiesen.

10.2 LIBPCAP

Die PCAP Bibliothek wurde für “tcpdump” entwickelt, um Pakete direkt von der Netzwerkkarte lesen zu können. Es wird dabei von jedem auf der Netzwerkkarte empfangenen oder zu sendenden Paket eine Kopie gemacht, so dass alle Pakete gesehen werden und dabei der Datenstrom von anderen Prozessen nicht beeinflusst wird.

Eine Netzwerkkarte empfängt normalerweise nur Broadcast- oder an sie adressierte Pakete. Wahlweise kann sie aber im “promiscuous mode” betrieben werden. Es würden dann unabhängig von der “Destination-Address” sämtliche Pakete, die bei der Netzwerkkarte vorbeikommen, empfangen und ans Betriebssystem weitergegeben. “Sniffer”-Programme, wie “tcpdump” oder “ethereal”, laufen defaultmässig im “promiscuous mode”, damit alle Pakete im Netzwerk von PCAP gesehen werden. Da die Pakete jedoch auch ans Betriebssystem weitergegeben werden, bedeutet dies eine zusätzliche Belastung für dieses.

Anschliessend folgt eine Auflistung und Erläuterung des Funktionsumfanges von PCAP, der gebraucht wurde, um den “Forwarder” zu realisieren:

pcap_t

Der “PCAP-Handle” “pcap_t” ist ein “typedef” der Struktur “struct pcap”. Sie repräsentiert einen “Device-Handle” der Netzwerkkarte, von der PCAP Pakete liest. Darin enthalten sind unter anderem ein Socket-Deskriptor (siehe pcap_open_live), ein Buffer für die eingelesenen Daten, ein Filter-Programm und ein Error-Buffer. Obwohl der direkte Zugriff auf diese Felder möglich ist, sollte er nicht durchgeführt werden. Es existieren Funktionen, die diese Aufgabe abstahieren.

pcap_open_live

Diese Funktion initialisiert den “PCAP-Handle” für eine “Device”. Dazu wird ein “Raw-Socket” erzeugt, das mit der Funktion “bind” direkt an diese Netzwerkkarte gebunden wird. Anhand dieses Sockets können Pakete direkt von der Netzwerkkarte gelesen oder geschrieben werden.

pcap_close

Gibt allozierten Speicher wieder frei und schliesst das Socket im “PCAP-Handle”.

pcap_lookupnet

Stellt die IP-Adresse eine Netzwerkkarte fest und gibt 0 zurück, falls dies möglich war und -1 falls nicht. Somit lässt sich feststellen, ob die Netzwerkkarte, auf der man horchen will, überhaupt aktiviert ist.

pcap_fileno

Gibt den im “PCAP-Handle” enthaltenen Socket-Deskriptor zurück, um anschliessend mit “write” oder “sendto” darauf schreiben zu können

pcap_read

Liest ein Paket aus dem Socket des “PCAP-Handles” ein und übergibt es einer “Callback”-Funktion, die beim Aufruf von “pcap_read” angegeben wurde. Die “Callback”-Funktion kann beliebig definiert werden, muss aber die Signatur von “pcap_handler” haben. Es ist somit möglich, ankommende Pakete zu analysieren und an den Aufrufer von “pcap_read” zurückzugeben.

11 Software-Modul *Verification*

11.1 Externe Beschreibung

11.1.1 Funktionsweise

Dieses Modul bildet das erste und zugleich letzte Glied in der Verarbeitung von Paketen des IP-Stacks auf dem “Mobile Host” und auf der “Base Station” für die Empfangs- beziehungsweise Senderichtung. Damit wird gewährleistet, dass alle Pakete welche je die Wireless-Karte verlassen beziehungsweise von dieser empfangen werden, zuerst durch dieses Modul geschickt werden. Zu diesem Zweck wird dieses Modul als Netfilter-Kernelmodul geladen und an die Hooks NF_IP_PRE_ROUTING sowie NF_IP_POST-_ROUTING gehängt.

Die gleichen Source-Dateien dieses Modules werden benutzt, um sowohl die “Mobile Host”- als auch die “Base Station”-Variante zu erzeugen. Herzu wird im Makefile die Auskommentierung der entsprechenden “Target”-Zeile entfernt.

Paketauthentisierung

In Senderichtung werden die Pakete mit einem IP-Header sowie dem “Verification Header” ergänzt. Das “Protocol Field” vom IP-Header trägt die freidefinierte Nummer 150[1]. Dabei wird im “Verification Header” der schon beschriebene MAC eingefüllt sowie ein Timestamp vermerkt. Das Originalpaket wird als Payload hinter den “Verification Header” gehängt und im “Next Field” des “Verification Header” mit der Protokollnummer 255 eingetragen, welches als “Raw Packet” gilt.

Auf der Gegenstation wird das Paket empfangen und der “Verification Header” untersucht. Falls die Zeitdifferenz gebildet aus Paket-Timestamp und lokalem Timestamp eine vorgegebene Schwelle überschreitet oder der MAC auf der Gegenstation nachgebildet nicht mit dem Paket-MAC übereinstimmt, so wird das Paket weggeworfen. Ansonsten gilt das Paket als authentisch und wird damit dem Protokoll-Stack übergeben.

Zeit-Synchronisierung

Die zweite Aufgabe dieses Modules besteht in der Synchronisierung des “Mobile Host” mit der Referenz-Zeit auf der “Base Station”, damit der Timestamp wegen eines “Time Shifts” nicht plötzlich abgelehnt werden.

[1] “Assigned Numbers”, RFC 1700; freier Bereich ist 101-254.

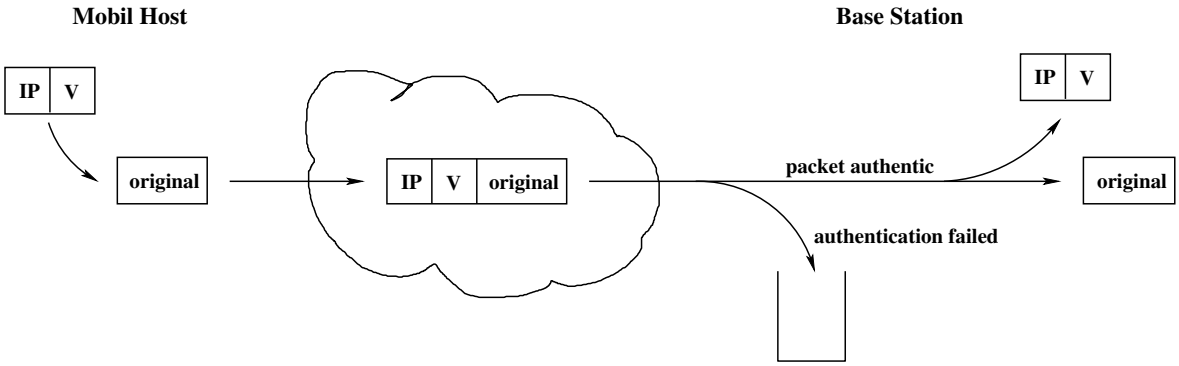


Abbildung 11.1: Sicherung der Wireless-Strecke

Im Konzept ist vorgesehen, dass erst wenn kein Datenverkehr stattfindet, der “Mobile Host” regelmässig den neuen Timestamp von der “Base Station” bezieht. Aus Vereinfachungsgründen wird in der Implementation unabhängig vom Datenverkehr alle 10 Sekunden ein “Timestamp Update” angefordert.

Der Vorgang wird ausgelöst sobald der “Mobile Host” ein “Time Update Request” abschickt. Dabei handelt es sich um ein Paket, welches nur aus dem IP-Header und dem “Verification Header” besteht und somit keine Nutzdaten trägt. Die Signalisierung findet im “Verification Header” statt, wo das Flag “Timestamp Packet” gesetzt wird. Damit erkennt dasselbe Modul auf der “Base Station”, dass eine “Timestamp Update Response” geschickt werden muss. Auf der “Base Station” wird nun ebenfalls ein Paket ohne Nutzdaten erzeugt und das Flag "Timestamp Packet" zwecks Signalisierung gesetzt.

Die Signalisierung direkt auf der “Verification Header”-Ebene wurde wegen des Aspekts der Bandbreitenschonung gewählt.

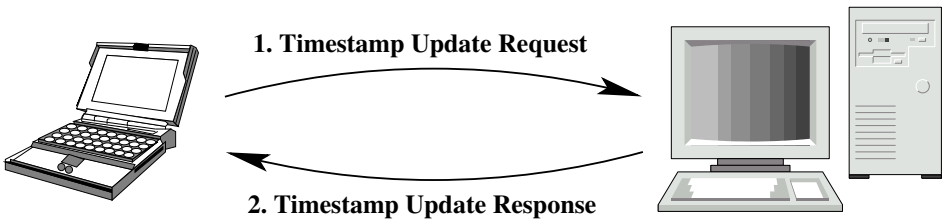


Abbildung 11.2: Timestamp Paketaustausch

11.2 Interne Beschreibung

11.2.1 Implementation

Die Hauptdatei in diesem Software-Modul ist “verification.c”. Dieses enthält die Initialisierungs- und Deinitialisierungsfunktionen, um als Kernel-Modul geladen zu werden.

Die Paketaufbereitung für das Anhängen des IP- sowie des “Verification Headers” werden in “IPHeader.h” und “VHeader.h” zur Verfügung gestellt. In “VHeader.c” befindet

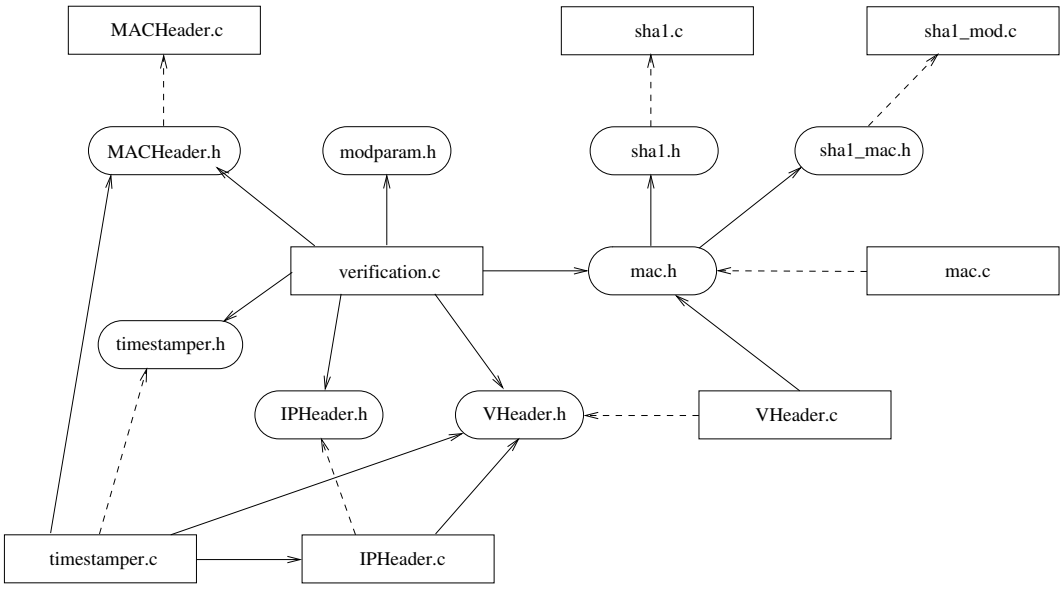


Abbildung 11.3: Datei-Abhängigkeiten

sich die Definition des “Verification Headers”, welcher aber über “VHeader.h” eingebunden werden kann.

In Abbildung 11.4 ist die Definition des “Verification Headers” abgebildet.

Next Header: Spezifiziert den Header oder das Protokoll welches anschliessend an den “Verification Header” folgt.

MAC Length: Gibt die Grösse des MACs in Anzahl Zeilen an.

Flags: 0000’0abc
a = “Timestamp Packet”
b = “Key Change Phase”
c = “Using New PID”

Reserved: Reserviert für späteren Gebrauch.

Timestamp: Hier wird ein Timestamp abgelegt, welcher einem Linux “struct timeval” in “Little Endian” entspricht.

MAC: Der “Message Authentication Code” trägt die Authentifikationsinformation des gesamten Pakets.

Die “mac”-Bibliothek wird durch “mac.c” implementiert und stellt Funktionen bereit sowohl für die Berechnung als auch das direkte setzen der PID. Im weiteren wird hier der MAC erzeugt.

Zur Zeit wird nur der SHA-1 Hash-Algorithmus unterstützt, welcher durch “sha1.h” zur Verfügung gestellt wird. Diese liegt in Bibliotheksform vor und wurde ursprünglich von

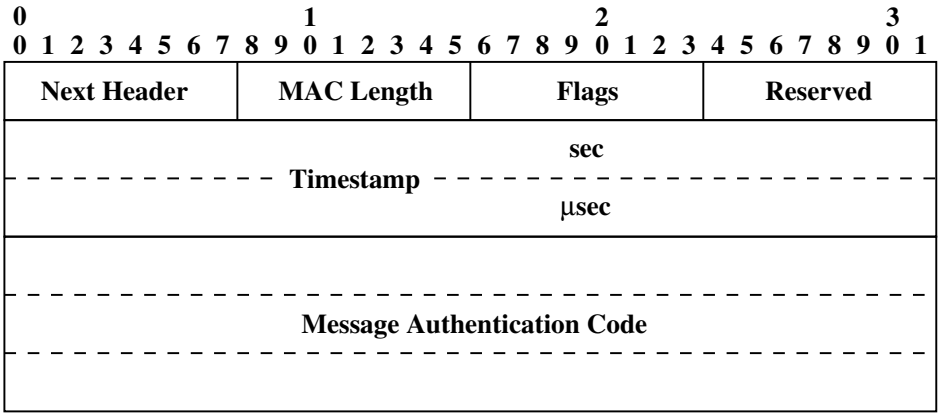


Abbildung 11.4: Definition von “Verification Header”

Steve Reid als “Public Domain” publiziert. Im “Verification”-Modul wird eine Kernel angepasste Version dieser Bibliothek von FreeSWAN benutzt.

Die “Keyed Hash Function” auf SHA-1 Basis wird durch “sha1_mac.h” zur Verfügung gestellt. Sie muss mit Hilfe von `sha1_mac_init()` und des gewünschten Schlüssels zuerst initialisiert werden. Damit werden die Initialisierungsvektoren des “inner & outer paddings” vorbereitet. Der MAC kann nun wiederholt mit dem Aufruf von `sha1_mac_calc()` generiert werden. Zum Schluss muss die Bibliothek mit `sha1_mac_finish()` deinitialisiert werden.

Paketauthentisierung

Das Hinzufügen und Entfernen des IP- und “Verification Headers” geschieht im `out_intercept_hook()` beziehungsweise `in_intercept_hook()`, welche sich in “verification.c” befinden. Im `struct nf_hook_ops`, welcher mit `nf_register_hook()` die beiden Funktionen registriert, wird als letzter Parameter die Priorität angeben. Hiermit kann die relative Reihenfolge der Paketverarbeitung gesteuert werden, falls mehrere Funktionen amselben Hook hängen. Für die “out”-Funktion wurde INT_MAX gewählt, um sicherzustellen, dass das Tunneln des Originalpakets als letztes geschieht. Die “in”-Funktion wird mit Priorität 0 aufgehängt, um zu garantieren, dass es als erstes die Gelegenheit bekommt, den “Verification Header” zu überprüfen.

“Mobile Host”-Variante:

Auf dem “Mobile Host” wird die PID an der Kommandozeile übergeben, weshalb die Funktion `calculate_pid()` nicht in diese Variante kompiliert wird. Die PID-Grösse wird von der Eingabelänge abgeleitet.

“Base Station”-Variante:

Die “Base Station” verfügt über die Funktion `calculate_pid()`, um die “Mobile Host” PID nachzubilden. Diese Funktion wird jedesmal im “in & out hook” aufgerufen. Alternativ könnten diese PIDs in einer Hash-Tabelle mit “Mobile Host”-Adressen als Index verwendet werden. Dies würde einen Performance-Gewinn erlauben, weil ein “Mobile

Host" auch bei häufigen Handoffs mehr als ein Paket zur gleichen "Base Station" schickt. Mit einem Timer könnten nicht mehr benötigte Einträge entfernt werden.

Zeit-Synchronisierung

Die Funktionalität für die Timestamp Synchronisierung befindet sich in "timestamper.c". Bevor dieser Mechanismus aktiv wird, muss die Implementation mit `init_timestamper_module()` initialisiert werden und bei Beendigung mit `free_timestamper_module()` wieder freigegeben werden.

"Mobile Host"-Variante:

Auf dem "Mobile Host" wird defaultmässig alle 10 Sekunden ein "Timestamp Update Request"-Paket erzeugt. Dieser Wert wird in "verification.c" `der init_timestamper_module()`-Funktion übergeben.

"Base Station"-Variante:

Nachdem die "Base Station" das "Timestamp Update Request"-Paket empfangen hat, wird defaultmässig nach einer Sekunde ein "Timestamp Update Response"-Paket erzeugt. Der Grund für diese Verzögerung ist, dass damit verhindert werden kann, dass wenn eine Flut von Request-Paketen eintreffen, diese mit einer Flut von Response-Paketen beantwortet werden. Dieser Wert wird ebenfalls in "verification.c" `der init_timestamper_module()`-Funktion übergeben.

11.2.2 Test

"Keyed Hash"-Bibliothek

Da die "Keyed Hash"-Bibliothek "sha1_mac" selbst implementiert wurde, muss eine ordnungsgemäss Arbeitsweise sichergestellt werden. Da nun jedoch keine Testwerte zur Verfügung standen, wurde die Bibliothek "mhash-0.8.9"[2] verwendet, um Testvektoren zu generieren. Weil keinerlei Beweise für die korrekte oder falsche Arbeitsweise dieser Bibliothek exisitieren, wurden diese Testvektoren mit einer Eigenimplementation der "Keyed Hash Function" und der "sha1.h"-Bibliothek verifiziert. Diese Testvektoren ("sha1_mac_test_vector.h") wurden schliesslich benutzt, um einen Test-Treiber ("sha1_mac_test_driver.c") zu schreiben, welcher die "sha1_mac"-Bibliothek überprüft.

Es gilt zu bemerken, dass die "Keyed Hash Function" in der "sha1_mac"-Bibliothek nicht auf dem Code basiert, welcher benutzt wurde, um die Testvektoren zu verifizieren.

Headermanipulationen

Da für das Hinzufügen des "IP-& Verification Headers" Kernelspeicher alloziert wird, muss nachgewiesen werden, dass keine "Memory Leaks" auftauchen. Im ersten Belastungstests welcher über die Nacht durchgeführt wurde, hängte sich die "Base Station" nach zwei Stunden auf. Nachforschungen ergaben, dass in der `calculate_pid()`

[2] Diese Bibliothek wurde in der Feasability-Phase untersucht. Sie konnte nicht als "Keyed Hash Function"-Lieferant benutzt werden, weil sie eine eigene Speicherverwaltung mitbringt und dies im Kernelspace problematisch ist.

Speicher alloziert aber nicht mehr freigegegeben wurde. Nach Behebung lief der Test erfolgreich.

Zeit-Synchronisierung

Dieser Teil des Modules wurde blackboxmässig mit Hilfe von “ethereal” getestet, indem überprüft wurde, ob alle 10 Sekunden ein “Time Update Request”-Paket an die “Base Station” geschickt wird und ob diese nach einer Sekunde mit einem “Time Update Response”-Paket antwortet.

In einer weiteren Phase wurden dann mit “ping” Datenpakete gesendet, womit überprüft werden konnte, ob das “Timestamp Packet”-Flag korrekt zurückgesetzt werden, unmittelbar nachdem Request- und Response-Pakete gesendet werden.

11.3 Probleme / Schwächen

Timestamp Genauigkeit

Es hat sich herausgestellt, dass das Erzeugen eines Timestamps obschon im Kernelspace mit Hilfe von get_fast_time() als sehr zeitabhängig von der gerade herrschenden Kernel-Belastung erwiesen hat. Es sollte deshalb untersucht werden, inwiefern der Timer der Wireless-Karte verwendet werden kann. Im Infrastruktur-Modus wird dieser Timer bei Abweichungen kontinuerlich durch das “Beacon Signal” nachjustiert. Da dies auf der Karte selbst geschieht ist die Genauigkeit höher und nicht von der Kernel-Last abhängig.

Initial Ping

Wegen der fehlenden Registrationsphase ergibt sich beim “Mobile Host” ein Henne-Ei-Problem: der Korrekturwert für den Timestamp kann nicht aus einem “Registration Response”-Paket abgeleitet werden. Um nun dennoch in den Vorteil des Timestamps als “Reply Attack”-Schutzmechanismus zu kommen, muss auf allen “Base Station” für jeden “Mobile Host” ein expliziter Ping durchgeführt werden. Dieser wird nur benötigt, um den initialen Korrekturwert nach dem Starten des “Mobile Hosts” zu bilden. Danach sorgt der obengenannte Timestamp-Synchronisierungs-Mechanismus für eine kontinuierliche Korrektur. Um diese Lösung zu automatisieren kann je ein Ping auf allen “Base Station” mit “Mobile Host”-Adresse im Intervall von 10 Sekunden gestartet werden[3].

[3] Es ist klar, dass hier unnötig Datenverkehr produziert wird. Ein Grosseinsatz des Netzes ist aber ohne Benutzerauthentisierung unwahrscheinlich, so dass sich der beschriebene Workaround für kleine Netze in Grenzen hält.

12 Software-Modul *Forwarder*

12.1 Externe Beschreibung

12.1.1 Funktionsweise

Aus der Sicht von unserer auf "Netfilter" basierten Lösung der Paket-Authentisierung besteht der grosse Nachteil der "Cellular IP" (→ CIP) Implementation der Columbia University darin, dass die Bibliothek "LIBPCAP" (→ PCAP) benutzt wird, um Pakete zu empfangen und zu senden. Um diese Aufgabe zu erledigen, interagiert PCAP direkt mit der Netzwerkkarte und umgeht somit den Protokoll-Stack und auch das gesamte "Netfilter"-Konzept von Linux (siehe Abbildung 12.1).

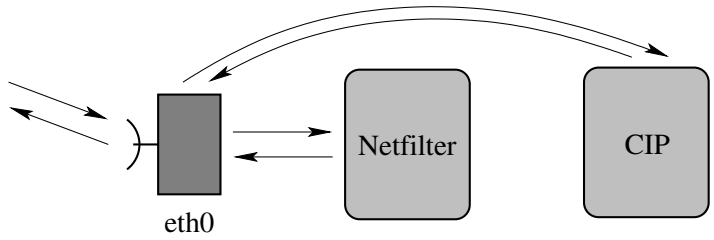


Abbildung 12.1: CIP umgeht Netfilter

Somit lassen sich mit "Netfilter" keine von CIP gesendeten Pakete authentisieren oder empfangenen Pakete verifizieren.

Als Lösung wurde daher der Umweg über das "Dummy-Device" entwickelt. CIP wird vorgegaukelt, dass das "Dummy-Device" das "Wireless-Device" ist, mit dem es kommunizieren soll. Mit Hilfe eines "Forwarders" können nun von CIP gesendete Pakete wiederum mit PCAP vom "Dummy-Device" gelesen und ans Betriebssystem weitergeleitet werden, wo sie im "Verification"-Modul mit dem "Verification Header" versehen werden. In der anderen Richtung werden auf der Netzwerkkarte empfangene Pakete vom "Verification"-Modul verifiziert und anschliessend vom "Forwarder" zum "Dummy-Device" gesendet, wo sie von CIP herausgelesen werden können (siehe Abbildung 12.2 auf der nächsten Seite).

Da Pakete, die das "Verification"-Modul passiert haben, für "Userspace"-Programme nicht zugänglich sind, musste ein "Netfilter"-Modul (Datei "forwarder_module.o") geschrieben werden, das solche Pakete zum "Dummy-Device" weiterleitet. Auf den "Mobile Hosts" werden keine ankommenden authentisierten Pakete an CIP weitergeleitet, da CIP nur das nicht-authentisierte "Beacon"-Signal empfängt. Datenpakete werden direkt

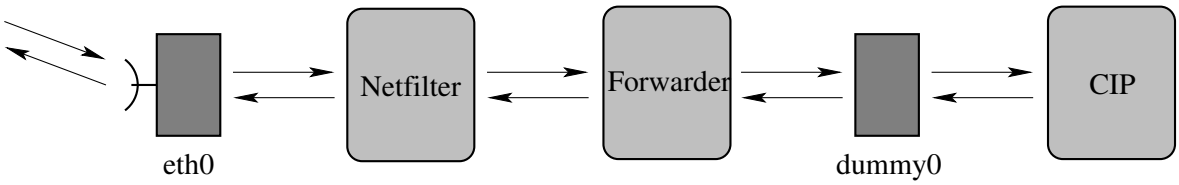


Abbildung 12.2: Lösung mit Dummy-Device

vom System an die betreffende Applikation gegeben. Somit wird dort das “Netfilter”-Modul nicht benötigt wird. Auf den “Base Stations” hingegen müssen alle auf dem “Wireless-Device” abgehenden Pakete authentisiert und ankommenden Pakete verifiziert werden.

CIP schreibt zu sendende Pakete direkt auf das “Dummy-Device”, so dass sie vom Linux-Kernel nicht gesehen werden. Die einzige Möglichkeit, solche Pakete dennoch ans “Verification”-Modul weiterleiten zu können, besteht darin, mit PCAP die Pakete vom “Dummy-Device” zu lesen. PCAP ist in einem Kernel-Modul nicht einsetzbar, so dass ein “Userspace”-Programm - ein “Forwarder-Daemon” (Datei “forwarder”) - geschrieben wurde, der diese Aufgabe vornimmt.

Der “Forwarder-Daemon” hat auf den “Base Stations” eine andere Funktionsweise, als auf den “Mobile Hosts”. Auf den “Base Stations” leitet er “Beacon”-Signale vom “Dummy-Device” direkt zum “Wireless-Device” und Daten-Pakete vom “Dummy-Device” zum “Verification”-Modul., wo sie schlussendlich auch zum “Wireless-Device” weitergegeben werden. Die Pakete werden vom “Forwarder-Daemon” somit nur in eine Richtung transportiert, bei den “Mobile Hosts” hingegen in beide Richtungen. Zum einen werden “Beacon”-Signale vom “Wireless-Device” zum “Dummy-Device” und zum anderen “Page”- und “Route-Update”-Pakete vom “Dummy-Device” zum “Wireless-Device” weitergeleitet.

12.2 Interne Beschreibung

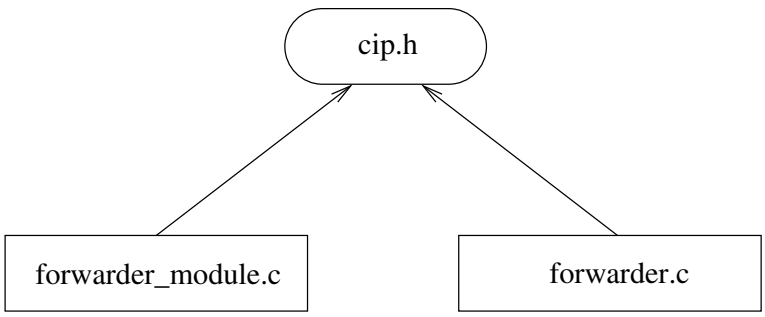


Abbildung 12.3: Datei-Abhängigkeiten

12.2.1 Implementation

Forwarder-Daemon (forwarder.c)

Beim Starten des “Forwarder-Daemons” wird zuerst ein Kontroll-Prozess gestartet, der zunächst versucht das “Lockfile” “/var/lock/forwarder” zu öffnen. Da nur ein Prozess auf einmal diese Datei öffnen und mit einem Schreibschutz versehen kann, ist sichergestellt, dass nur ein “Forwarder-Daemon” auf einmal laufen kann. Indem sich der Hauptprozess beendet, läuft der Kontroll-Prozess als Daemon im Hintergrund weiter. Dieser startet anschliessend für jede “Forwarder”-Richtung einen neuen Prozess, der die entsprechenden Pakete behandelt.

Auf den “Mobile Hosts” werden somit zwei neue Prozesse gestartet, wobei in dem einen die Funktion “net_to_dummy_forwarding” und im anderen die Funktion “dummy_to_net_forwarding” läuft. Sie übernehmen das Weiterleiten der Pakete in, indem sie in einer Endlos-Schleife Pakete von einem Device mit lesen und ins andere Schreiben. Zum Lesen der Pakete wird die Funktion “pcap_read” (siehe Kapitel 10.1 auf Seite 61) verwendet, der eine “Callback”-Funktion übergeben wird. Diese “Callback”-Funktion wird bei jedem empfangenen Paket aufgerufen, um zu entscheiden, ob das Paket weitergeleitet werden soll oder nicht. Wie in Kapitel 12.1.1 auf der vorherigen Seite beschrieben, werden somit einerseits die “Beacon”-Signale und andererseits die “Route”- und “Page-Update”-Pakete akzeptiert.

Auf den “Base Stations” wird hingegen nur ein Prozess gestartet, auf dem die Funktion “dummy_to_net_forwarding” läuft. In dieser Funktion werden in einer Endlosschleife die Pakete mit “pcap_read” vom “Dummy-Device” geholt und wie in Kapitel 12.1.1 auf der vorherigen Seite beschrieben, entweder direkt zum “Wireless-Device” oder übers Betriebssystem zum “Verification”-Modul weitergeleitet. Um die zu authentisierenden Pakete ans Betriebssytem weiterzugeben, wird ein “Raw-Socket” verwendet.

Forwarder-Modul (forwarder_module.c)

Die Netfilter-Implementierung des “Forwarders” besteht aus den Funktionen “fw_intercept_hook”, die an den Hook “NF_IP_FORWARD” und “out_intercept_hook”, die an den Hook “NF_IP_POST_ROUTING” angehängt ist (siehe Kapitel 10.1 auf Seite 61).

“fw_intercept_hook” wird aufgerufen, sobald ein Paket im System eintrifft. Es werden dabei folgende Entscheidungen getroffen:

- Falls das Packet nicht auf dem “Wireless Device” angekommen ist, so wird es weggeworfen. Somit ist sichergestellt, dass das Routing der Pakete nur von CIP durchgeführt wird.
- Ankommende “Beacon”-Signale werden ebenfalls weggeworfen. Sie werden von CIP auf der “Base Station” nicht benötigt.
- Alle übrigen Pakete werden zum “Dummy-Device” weitergeleitet.

Da diese Funktion am “hook” “NF_IP_FORWARD” angehängt ist, müssen an die “Base Station” adressierte Pakete nicht speziell behandelt werden, da sie diesen “hook” nie passieren. Sie werden stattdessen zum “hook” “NF_IP_LOCAL_IN” weitergeleitet.

Die Funktion “out_intercept_hook” hingegen wird aufgerufen, bevor ein Paket das System verlässt. Generell werden hier alle Pakete zum “Wireless-Device” weitergeleitet. Dies ist nötig, da Pakete, die von CIP zum “Dummy-Device” und von dort via “Forwarder-Daemon” ins System gelangt sind, dem Routing von Linux unterliegen und auf irgendein “Device” weitergeleitet werden können. Da für CIP das “Dummy-Device” das “Wireless-Device” repräsentiert, müssen diese Pakete das System auf dem letzteren verlassen.

Für die Administration wurde jedoch hier eine Hintertüre offen gelassen, indem Pakete, die als “Source Address” den Lokalen Rechner eingetragen haben, nicht umgeleitet werden. Dies ermöglicht es, eine “Base Station” “remote” zu administrieren.

12.2.2 Test

Forwarder-Daemon

Der “Forwarders-Daemon” wurde in Zusammenarbeit mit CIP getestet, indem ein “Mobile Host” gemäss Kapitel 14.4 auf Seite 85 jedoch ohne das “Verification”-Modul konfiguriert wurde. Das von der “Base Station” empfangene “Beacon”-Signal musst vom “Forwarder-Daemon” nun zum “Dummy-Device” weitergeleitet werden, wo sie mit “tcpdump” überprüft werden konnten.

In der Gegenrichtung sollte das “Page”- und “Route-Update”-Signal vom “Forwarder-Daemon” vom “Dummy-Device” gelesen und auf die Netzwerkkarte geschrieben werden. Auf der “Base Station” konnte das korrekte Empfangen der Pakete mit einem Sniffer-Programm überprüft werden.

Forwarder-Modul

Das “Forwarder”-Modul konnte auch zusammen mit CIP getestet werden. Es wurde auf einer “Base Station” anhand der Anleitung in Kapitel 14.3 auf Seite 83 jedoch ohne das “Verification”-Modul konfiguriert. Die ankommenden “Page”- und “Route-Update”-Pakete konnten auf dem “Dummy-Device” mit “tcpdump” überprüft werden.

Das “Forwarden” von Daten-Paketen liess sich mit dem Programm “ping” vom “Mobile Host” aus testen. Wenn ein externer Rechner “angepingt” wurde, so musste der “Mobile Host” die Response-Pakete erhalten, auf dem “Dummy-Device” der “Base Station” Request- und Response-Pakete zu sehen sein und keine Pakete übers Linux-Routing auf der “Base Station” weitergeleitet werden. Der letzte Punkt konnte überprüft werden, indem die “Routing-Table” so angepasst wurde, dass das Antwort-Paket nicht auf das “Wireless-Device” geroutet wurde. Indem auf dem neu eingetragenen “Device” mit einem “Sniffer”-Programm gehorcht wurde, konnte das Nicht-Erscheinen des Response-Paketes festgestellt werden.

12.3 Probleme / Schwächen

Doppeltes “Forwarding”

Pakete die mit Hilfe des “Forwarder”-Moduls vom “Wireless-Device” zum “Dummy-Device” gesendet wurden, sind dort vom “Forwarder-Daemon” wieder herausgelesen und übers Betriebssystem zurück zum “Wireless-Device” geschickt worden. Da der “Forwarder-Daemon” Daten-Pakete von beide Richtungen auf dem “Dummy-Device” sieht, wusste er nicht, ob er ein Daten-Paket weiterleiten sollte oder nicht.

Als Lösung wird das reservierte Bit 0 der Flags des IP-Headers verwendet, um ein Paket zu kennzeichnen, das vom “Forwarder”-Modul zum “Dummy-Device” geleitet wurde. Der “Forwarder-Daemon” ignoriert nun alle Pakete, die dieses Bit gesetzt haben.

13 Integration mit Cellular IP v1.1

13.1 Vorgehensweise

Die Integration des “Verification”-Moduls mit der Referenzimplementation erfolgte mit Hilfe des “Forwarders” anhand der in Kapitel 14.3 auf Seite 83 und Kapitel 14.4 auf Seite 85 beschriebenen Konfigurationen.

13.2 Integrationstest

Um die Leistungsfähigkeit der verschiedenen Software-Module testen zu können, wurde ein Testnetz gemäss Abbildung 13.1 aufgebaut.

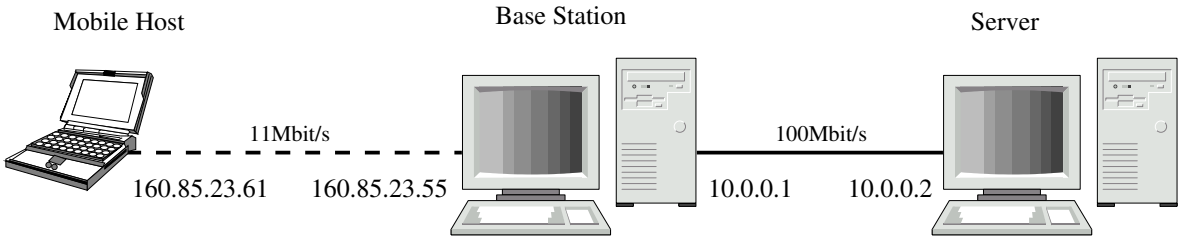


Abbildung 13.1: Integrations-Testnetz

Ein Testreihe bestand aus dem fünfmaligen Herunterladen des Linux Kernels 2.4.5 (25.3MB) vom Server und dem Bilden des Mittelwertes der benötigten Zeit und des Durchsatzes. Anschliessend folgt eine Auflistung der durchgeführten Testdurchgänge für verschiedene Konstellationen. Als Referenz wurde auch noch eine Testreihe von der “Base Station” aus ausgeführt.

Beschreibung	Zeit [s]	Durchsatz [kbit/s]
BS: Normaler Betrieb, ohne CIP, Verification, Forwarder	29	878.8
MH: Normaler Betrieb, ohne CIP, Verification, Forwarder	140	183.6
MH: CIP, Verification, Forwarder	158	164.0
MH: Nur Verifcation	145	178.0

13.3 Probleme / Schwächen

Modul entladen

Bei der Integration des “Verification”-Moduls auf einer “Base Station” stellte sich heraus, dass sich Linux nicht mehr korrekt herunterfahren liess. Der Grund dafür liegt darin, dass für das “Wireless-Devices” der “Reference-Counter” einen zu hohen Wert angibt. Das heisst, dass der Treiber des “Wireless-Devices” noch von einem anderen Modul gebraucht wird und daher nicht freigegeben werden kann. Intensive Nachforschungen haben ergeben, das der Zugriff auf die ARP-Lookup-Tabelle diesen Counter einmalig erhöht.

Dieser Fehler wurde nur deshalb erst bei der Integration entdeckt, weil das “Verification”-Modul auf den normalen Netzwerkkarten einwandfrei funktionierte. Der naheliegendste Grund für das Auftreten des Problemes ist ein Fehler im Gerätetreiber des “Wireless Devices”.

Teil V

Anwendung

14 Installation und Konfiguration

14.1 Kernel

Es wird ein Kernel benötigt, der “Netfilter” unterstützt - empfohlen ist mindestens Version 2.4.4. Für das “Verification”-Modul und den “Forwarder” sind, wie nachfolgend für den Kernel 2.4.5 beschrieben, verschiedene Einstellungen vorzunehmen.

Verification-Modul

Unter “Network options” ist die Option “Network packet filtering (replaces ipchains)” anzuwählen. Die Option “Fast switching (read help!)” darf *nicht* aktiviert werden. Die Einstellungen unter “Netfilter Configuration” werden nicht benötigt, können aber, falls “IP-Tables” eingesetzt werden soll, aktiviert werden.

Forwarder

Unter “Network options” sind die Optionen “Packet Socket” und “Socket Filtering” anzuwählen (werden nur für PCAP benötigt). Auf den "Base Stations" wird zudem noch "Network packet filtering (replaces ipchains)" aus demselben Menü benötigt. Die Option “Fast switching (read help!)” darf *nicht* aktiviert werden.

Unter “Network device support” muss die Option “Dummy net driver support” als Modul angewählt werden.

14.2 Orinoco Wireless PC-Card

Die Einstellungen für diese Netzwerkkarte wurden grundsätzlich von der Vorgängerarbeit “Wireless LAN basierend auf Cellular IP” von Marc Steiner und Thomas Wendel übernommen. Hier sollen nur die Änderungen aufgezeigt werden, die bei der Umstellung zum 2.4er Kernel vorgenommen wurden.

Die Orinoco Wireless PC-Cards werden im 2.4er Kernel mit dem Treiber “orinoco_cs” unterstützt. Im Kernel Version 2.4.5 sind dazu folgende Einstellungen nötig:

- Unter “General setup” müssen im Menü “PCMCIA/CardBus support” die Optionen “PCMCIA/CardBus support”, “CardBus support” und “i83265 compatible bridge support” aktiviert werden.

- Unter “Network device support” sind im Menü “PCMCIA network device support” die Optionen “Pcmcia Wireless LAN” und “AT&T/Lucent Wavelan wireless support” anzuwählen.
- Auch unter “Network device support” ist ist im Menü “Wireless LAN (non-hamradio)” die Option “Wireless LAN (non-hamradio” und “Hermes support (Orinoco/Wavelan-IEEE/PrismII/Symbol 802.11b cards)” zu aktivieren.

Wie sich herausgestellt hat, wird ein Beacon-Signal vom Orinoco-Treiber fälschlicherweise als SNAP in einem 802.2 LLC Paket in einem 802.11 Frame erkannt. Dies führt dazu, dass der SNAP-Header entfernt, das Ethernet II Frame rekonstruiert und die Original-Daten überschrieben werden. Ein solche Paket wird von CIP nicht mehr als Beacon-Signal erkannt und somit sieht ein Mobile Host keine "Base Station" mehr.

Als Lösung wurde der Orinoco-Treiber so modifiziert, dass die Überprüfung auf ein solches SNAP-Frame nicht mehr stattfindet. Beim Kernel 2.4.5 bedeutet dies, dass in “drivers/net/wireless/orinoco.c” die Zeilen 1107-1109

```
if(((status&HERMES_RXSTAT_MSGTYPE)==HERMES_RXSTAT_1042) ||
  ((status&HERMES_RXSTAT_MSGTYPE)==HERMES_RXSTAT_TUNNEL) ||
  (!memcmp(&hdr.p8022, &encaps_hdr, 3))) {
```

geändert werden zu

```
if(((status&HERMES_RXSTAT_MSGTYPE)==HERMES_RXSTAT_1042) ||
  ((status&HERMES_RXSTAT_MSGTYPE)==HERMES_RXSTAT_TUNNEL)) {
```

Diese Änderungen müssen durchgeführt werden, bevor der Kernel kompiliert wird.

Im Verzeichnis “/etc/pcmcia” muss noch der neue Treiber anschliessend eingetragen werden:

- In “config” ist der Eintrag

  ```
  device "orinoco_cs"
    class "network" module "orinoco_cs"
  ```

  einzufügen und die letzte Zeile des Eintrags

  ```
  card "Lucent Technologies WaveLAN/IEEE Adapter"
  version "Lucent Technologies", "WaveLAN/IEEE"
  bind "wvlan_cs"
  ```

  zu ersetzen durch

  ```
  bind "orinoco_cs"
  ```

- In “config.opts” ist der Eintrag

```
module "orinoco_cs" opts "irq_list=11"
```

einzufügen.

Die Orinoco Wireless PC-Card wird beim Booten nach den normalen Netzwerkkarten als weiteres eth-Device (z.B. eth2) erkannt.

ACHTUNG: Aus unbekannten Gründen werden die in “wireless.opts” eingetragenen Einstellungen nicht übernommen, so dass entweder von Hand oder in einem Skript “iwconfig eth2 essid "TEST" mode ad-hoc rate 11M” ausgeführt werden muss.

14.3 Base Station

Um eine “Base Station” in Betrieb zu nehmen, müssen der Reihe nach das “Dummy-Device” eingerichtet, das “Verification-Modul” geladen, der “Forwarder” und schliesslich “CIP” gestartet werden. Die Konfiguration und Installation der verschiedenen Teile wird nachfolgend erklärt.

Dummy-Device

CIP sendet seine Packete ans "Dummy-Device" und somit wird dessen Adresse als Absender ins Paket eingetragen. Um nicht nachträglich diese Adresse auf diejenige des "Wireless-Devices" ändern zu müssen, wird das "Dummy-Device" direkt mit der Adresse des "Wireless-Devices" konfiguriert. Dazu müssen folgende Schritte vorgenommen werden:

1. Feststellen der IP- und MAC-Adresse des "Wireless-Devices":

   ```
   > ifconfig eth2
   eth3 Link encap:Ethernet HWaddr 00:02:2D:05:7D:1B
   inet addr:160.85.23.55 Bcast:160.85.23.63 ...
   ```

2. Falls das "Dummy-Device" schon benutzt wird, muss es deaktiviert werden:

   ```
   > ifconfig dummy0 down
   ```

3. Als nächstes wird die MAC-Adresse gesetzt:

   ```
   > ifconfig dummy0 hw ether 00:02:2D:05:7D:1B
   ```

4. Anschliessend erfolgt das Setzen der IP-Adresse und Aktivieren des Dummy-Devices:

   ```
   > ifconfig dummy0 160.85.23.55 up
   ```

5. Aus der Routing-Tabelle muss die Route übers "Dummy-Device" herausgelöscht werden:

```
> route -n
Destination  Gateway  Genmask       ...  Iface
160.85.0.0   0.0.0.0  255.255.0.0   ...  dummy0
...
> route del -net 160.85.0.0/16 dummy0
```

CIP v1.1

Auf den "Base Stations" wurden die Einstellungen der Vorgängerarbeit "Wireless LAN basierend auf Cellular IP" in der Datei "pa_basestation_dedicated.conf" übernommen. Lediglich das "Wireless-Device" wurde von "wvlan0" auf "eth2" geändert.

In der Datei "cipnode.c" wurde in der Funktion "create_beacon" das Längenfeld des "Beacon"-Signals (Bytes 12-13) korrigiert. Fälschlicherweise wurde eine Länge von 55 fest eingetragen, obwohl die Gesamtlänge des "Beacon"-Signals 61 Bytes beträgt. Dies führte dazu, dass die mitgelieferte Gateway-IP-Adresse am Ende des Paketes abgeschnitten wurde. Diese wird jedoch von den "Mobile Hosts" als Destination-Adresse für "Page-Update"- oder "Route-Update"-Pakete gebraucht.

Um "cipnode" zu übersetzen muss im "Makefile" beim "Target" "cipnode.o" die Makros "ARPFILTER" und "FILTER" definiert und anschliessend durch Eingeben von "make" generiert werden. "ARPFILTER" ist nötig, damit ARP-Pakete nicht zum nächsten Knoten weitergeleitet werden. Mit "FILTER" wird verhindert, dass CIP Pakete in einer Endlosschleife zwischen dem "Uplink"- und dem "Downlink-Device" hin- und herroutet.

Abschliessend kann "cipnode" gestartet und im Hintergrund laufen gelassen werden:

```
> nohup ./cipnode pa_basestation_dedicated.conf > \
  cipnode.log &
```

Verification-Modul

Um das "Verification"-Modul zu übersetzen, muss im "Makefile" die Zeile "TARGET=__BASE_STATION__" eingetragen und allfällig die Zeile "TARGET=__MOBILE_STATION__" entfernt werden. Durch das Eingeben von "make" wird das Modul "vmod_bs.o" erzeugt.

Um es zu laden wird der Netzwerkschlüssel benötigt. Er ist momentan noch frei wählbar und statisch, wird zu einem späteren Zeitpunkt jedoch vom "Gateway" periodisch generiert und verteilt. Zusätzlich muss dem "Verification"-Modul angegeben werden, welches das "Wireless-Device" ist:

```
> insmod  vmod_bs.o  air_dev="eth2"  net_key="0102030405"
```

Forwarder

Um den "Forwarder" mit "make all" übersetzen zu können muss ebenfalls im "Makefile" die Zeile "TARGET=__BASE_STATION__" eingetragen und allfällig die Zeile "TARGET=__MOBILE_STATION__" entfernt werden. Es werden die beiden Dateien "forwarder" und "forwarder_module.o" erzeugt.

Den beiden Forwardern muss beim Starten jeweils die Netzwerkkarte und das “Dummy-Device” angegeben werden, zwischen denen sie “forwarden” sollen:

```
> forwarder --net eth2 --dummy dummy0
> insmod forwarder_module.o net_dev="eth2" \
  dummy_dev="dummy0"
```

Die weiteren Konfigurationsmöglichkeiten des “Forwarders” können, wie folgt gezeigt, gewählt werden:

```
> forwarder --help
Usage: forwarder parameters [options]
Parameters:
-d, --dummy <dummy_device>  Specifies the dummy device
-n, --net <net_device>      Specifies the net device
Options:
-o, --nodaemon              Don't run forwarder as daemon
-v, --verbose               Display read packets
--help                      Display this information
```

Zeit-Synchronizität

Wie in Abschnitt 7.3.6 auf Seite 49 aufgezeigt wird, müssen die Uhren der “Base Stations” synchron laufen. Wie dort erwähnt, kann für diese Problemstellung das “Network Time Protocol”[1] als Lösung verwendet werden.

14.4 Mobile Host

Dummy-Device

Das “Dummy-Device wird gleich wie in Kapitel 14.3 auf Seite 83 beschrieben konfiguriert.

CIP v1.1

Auf den “Mobile Hosts” wurde in der Datei “ cipmobile.conf” das “Wireless-Device” von “wvlan0” auf “dummy0” geändert. Alle anderen Einstellungen wurden von der Vorgängerarbeit übernommen.

Nach der Übersetzung des Programms mit “make” kann es gestartet werden:

```
> ./cip
```

[1] http://www.eecis.udel.edu/~ntp/

Verification-Modul

Um das “Verification”-Modul zu übersetzen, muss im “Makefile” die Zeile “TARGET=__MOBILE_STATION__” eingetragen und allfällig die Zeile “TARGET=__BASE_STATION__” entfernt werden. Es wird dabei das Kernel-Modul “vmod_mh.o” erzeugt.

Zu einem späteren Zeitpunkt wird ein“Mobile Host” seine “PID” während der Registrationsphase von der “Base Station” enthalten. Da dieser Registrationsprozess jedoch noch nicht implementiert ist, muss die “PID” von Hand erzeugt und dem “Verification”-Modul beim Laden übergeben werden. Die “PID” wird aus der IP-Adresse des “Mobile Hosts” und dem Netzwerkschlüssel generiert.:

```
> ./calc_pid  0102030405  160.85.23.61
Network Key (5) = 0102030405
Generated PID (20) = d4ec046e4ed976e061f256541650ab5b0e684d8c
```

Mit der “PID” und der Angabe des “Wireless-Devices” und der Grösse der “MAC”, die im “Verifcation Header” enthalten ist, wird das “Verification”-Modul geladen:

```
> insmod vmod_mh.o air_dev="eth0" out_mac_size=12 \
  pid="d4ec046e4ed976e061f256541650ab5b0e684d8c"
```

Forwarder

Um den “Forwarder-Daemon” mit “make forwarder” übersetzen zu können muss im “Makefile” die Zeile “TARGET=__MOBILE_STATION__” eingetragen und allfällig die Zeile “TARGET=__BASE_STATION__” entfernt werden. Es wird das Programm “forwarder” erzeugt, das mit der Angabe der Netzwerkkarte und des “Dummy-Devices”, zwischen denen es “forwarden” soll, gestartet werden kann:

```
> forwarder --net eth2 --dummy dummy0
```

Zeit-Synchronizität

Wie in Abschnitt 32 auf Seite 65 beschrieben ist, müssen die “Mobile Hosts” und “Base Stations” ihre Pakete mit der selben Zeit-Basis authentisieren. Damit ein “Mobile Host” den Korrekturwert für seine Zeit bilden kann, muss er vor dem Absenden des ersten Pakets die Referenz-Zeit der “Base Station” erhalten.

Eine mögliche Lösung ist das “Anpingen” des “Mobile Hosts” von der “Base Station” aus. Dieser wird sich sogleich auf das erste ankommende Paket synchronisieren.

Teil VI

Projektverlauf

15 Zeitplanung

Aus den nachfolgenden beiden Abbildung wird die Zeitplanung dieser Projektarbeit ersichtlich. Am Anfang stand nur eine grobe Zeitaufteilung fest, welche hauptsächlich aus der Aufgabenstellung abgeleitet wurde.

15.1 Projektverlauf

Die erste Woche wurde für die Einarbeitung verwendet. Hierbei haben sich beide Studenten in die Materie des “Cellular IP” eingelesen sowie den Source-Code der Referenz-Implementation studiert. Auch wurde ein erster Feasabililty-Test mit LIBPCAP unternommen, um Pakete zu empfangen.

In der zweiten Woche wurden intensive Feasability-Tests mit dem Netfilter-Mechanismus unternommen. Die Beispielprogramme, welche mit der Dokumentation mitgeliefert werden, lassen zwar sehr schnell einfache Kernelmodule mit Netfilter-Funktionalität schreiben, jedoch musste das Wissen um das Paketbuffer-Management für das Einfügen von zusätzlichen Headern mühsam erarbeitet werden, zumal der Kernel Fehler mit einem hängenden System bestraft hat.

Parallel wurde in dieser Woche begonnen, ein Konzept für die Benutzerauthentisierung zu erarbeiten. Wie in der Problemfeldanalyse aufgezeigt, weist einerseits der Standard Schwächen auf als auch die Referenz-Implementation selbst, so dass eine Erweiterung der Referenz-Implementation äussert von zweifelhaftem Ausgang gewesen wäre. Im Einverständis mit unserem Dozenten Herrn Dr. Steffen, wurde deshalb zuerst das Augenmerk auf eine saubere Konzepterstellung gelegt. Durch Hinzukommen von immer neuen Anforderungen einerseits durch die Konzeptanalyse selbst, als auch mit Hilfe von vorgeschlagenen Lösungsansätzen durch Herrn Dr. Steffen konnte das Konzept perfektioniert werden, jedoch wurde dabei auch der Anfangszeitplan überschritten.

Es zeichnete sich ab, dass die ursprüngliche Aufgabenstellung so nicht mehr erfüllt werden kann. Dies kann aber wie oben schon erwähnt mit der Problemfeldanalyse gerechtfertigt werden.

Um nun am Schluss der Projektarbeit dennoch einen Mehrwert aus Implementationssicht erzeugen zu können, wurde die Paketauthentisierung realisiert und mit der Referenz-Implementation integriert. Die Paketauthentisierung sichert somit den Normalbetrieb des WLANs ab, indem alle Datenpakete inklusive Kontrollpakete authentisiert übertragen werden.

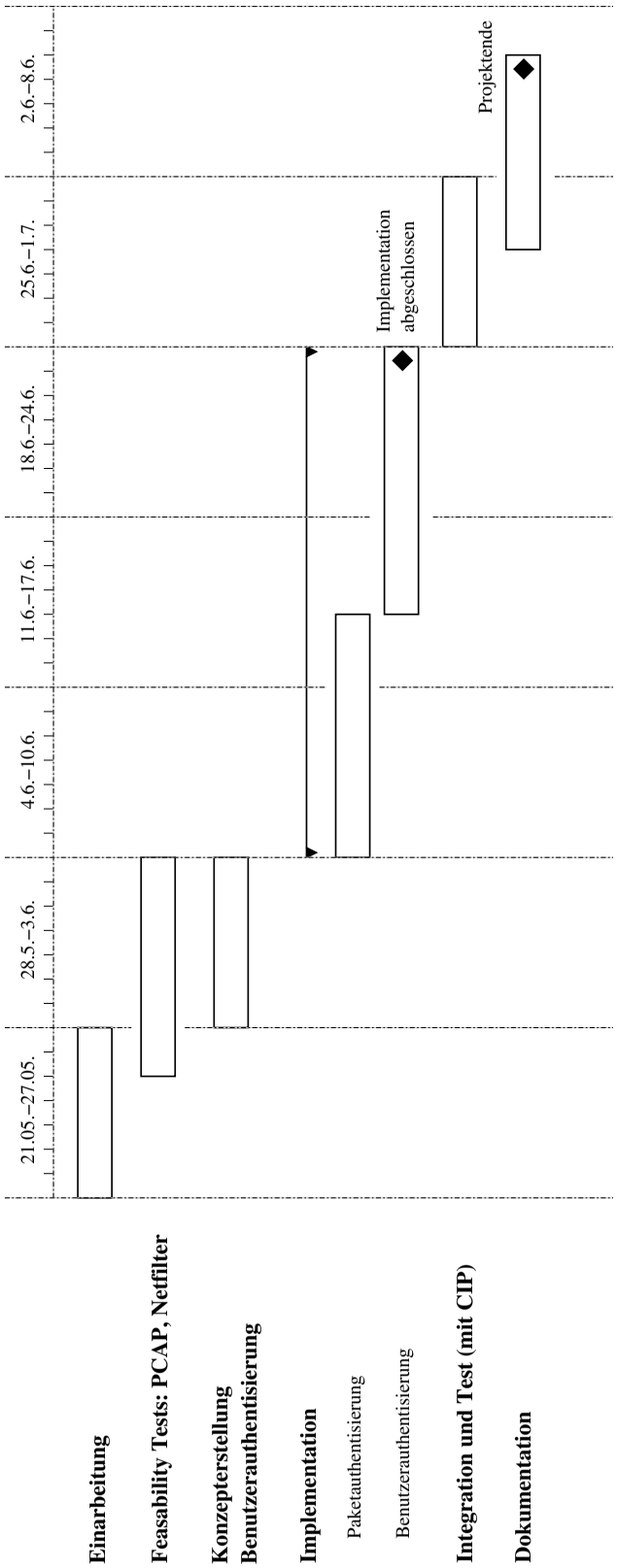


Abbildung 15.1: Zeitplanung zum Zeitpunkt 25.05.2001

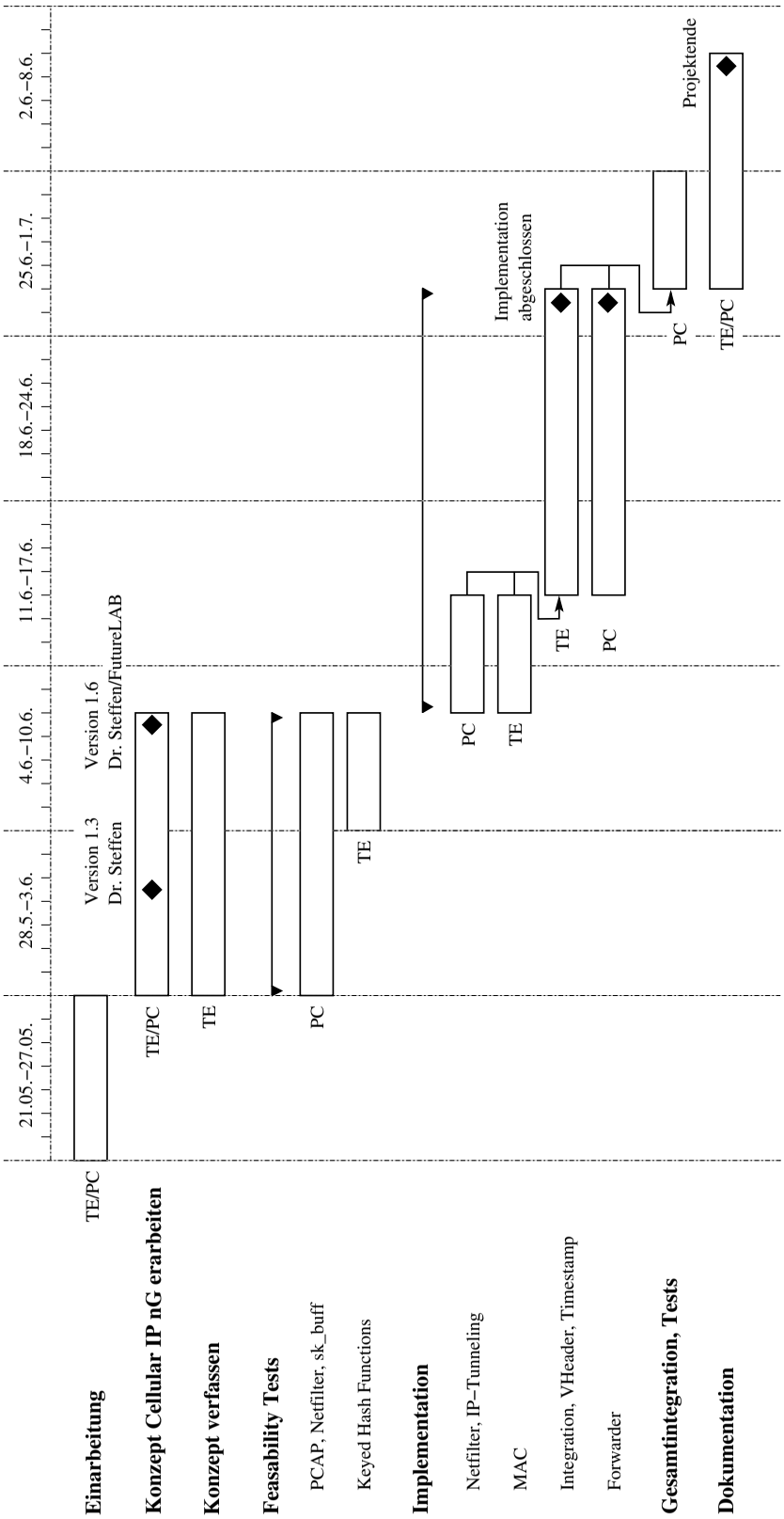


Abbildung 15.2: Zeitplanung zum Abgabezeitpunkt 06.07.2001

16 Schlussbemerkungen

16.1 Fazit

In einer ersten Phase wurde zuerst versucht eine Benutzerauthentisierung als Erweiterung für “Cellular IP” zu spezifizieren. Eine Problemfeldanalyse hat jedoch ergeben, dass dies nicht durchführbar ist, ohne Erweiterung des “Cellular IP”-Standards selbst. Deshalb wurde ein umfassendes Konzept fur ein grossflächiges WLAN enwickelt namens “Cellular IPnG”. Dieses Konzept erlaubt eine geschlossene Implementation, bleibt aber konzeptionell ausbaubar.

In einer zweiten Phase wurde die Paketauthentisierung aus dem Konzept herausgegriffen und implementiert. Um eine vollständige Integration mit der bestehenden Referenz-Implementation zu erreichen, wurden kreative Lösungen erarbeitet.

Aus Zeitgründen konnte die Benutzerauthentisierung aus dem Konzept nicht mehr implementiert werden. Jedoch wird in der realisierten Software darauf wertgelegt, dass sie auch im praktischen Umfeld eingesetzt werden kann.

Diese Projektarbeit hat uns die Möglichkeit gegeben, ein sehr weites Tätigkeitsfeld abzudecken. Angefangen von der theoretischen Herausforderung bei der Konzepterstellung, über die kreative Herausforderung der Lösungsfindung, wo eine Brücke zur Referenz-Implementation geschlagen werden musste, bis zur programmiertechnischen Herausforderung im Umgang mit der Kernelprogrammierung.

16.2 Ausblick

Grossflächige Wireless-LANs gehören mit ihren hohen Übertragungsraten zur übernächsten Generation von Mobilfunknetzen. “Cellular IP” bringt hierbei den Ansatz der Paketvermittlung über eine zellulare Infrastruktur ein, welche auf dem Standard 802.11 aufsetzt.

Das in dieser Projektarbeit erstellte Konzept löst wichtige Schwachpunkte von “Cellular IP”, welche für ein grossflächiges Wireless-LAN unabdingbare Eckpfeiler darstellen. Zu nennen sind hier beispielsweise die Benutzerauthentisierung, das Accounting sowie die Sicherung der Luftstrecke vom “Mobile Host” zur “Base Station”.

Das erstellte Konzept stellt nicht der Weisheit letzter Schluss dar. Auf konzeptioneller Ebene können weitere Accountingverfahren als Module hinzugenommen werden. Auch muss der Frage nachgegangen werden, wie IPv6 in das Konzept passt. Weil der “Cellular IP”-Standard 1999 ausgelaufen ist, stellt sich die Frage, ob das neue Konzept nicht

als Draft-Standard eingereicht werden soll. Interessant zu untersuchen wäre es, ob Neuronale Netzwerke angewendet werden könnten, um eine effekitve Lastverteilung zu erreichen. Dies ist deshalb interessant, weil zu erwarten ist, dass sich die Bewegungen im “Cellular IP”-Netz geprägt sein werden durch die Gewohnheiten der Teilnehmer.

Auf der Implementationsseite ist zu erwarten, dass die Wireless-Karten im Infrastruktur-Modus betrieben werden und somit ihren vollen Leistungsumfang ausschöpfen können. Auch ist zu erwarten, dass die instabile Referenz-Implementation der Columbia University durch eine neue, kernelspaceorientierte Lösung ersetzt werden wird. Hierzu wird höchstwahrscheinlich das Netfilter-Konzept zum Zuge kommen. Um die Sicherheit eines grossflächigen WLANs nicht zu kompromittieren, müssen auch Lösungen bezüglich “Intrusion Detection” erarbeitet werden.

Unsere Projektarbeit kann als Ausgangslage verwendet werden, um in einem ersten Schritt eine Neuimplementation des Routing-Mechanismus von “Cellular IP” auf der einen Seite, sowie die Umsetzung der im Konzept erarbeiteten Registrierungsphase auf der anderen Seite, durchzuführen. In einem weiteren Schritt kann dann das Accounting mit der “Cryptocard”-Lösung hinzugenommen werden. Auch ist eine Implementation des Konzeptes auf die Windowsplattform insbesondere des WindowsCE Betriebssystems für einen kommerziellen Durchbruch unabdingbar.

Auf der Infrastrukturseite könnte die gesamte ZHW vernetzt werden, um sowohl ein Forschungsnetz als auch ein operatives Netz zu erhalten, welches von Studenten und Dozenten genutzt werden könnten.

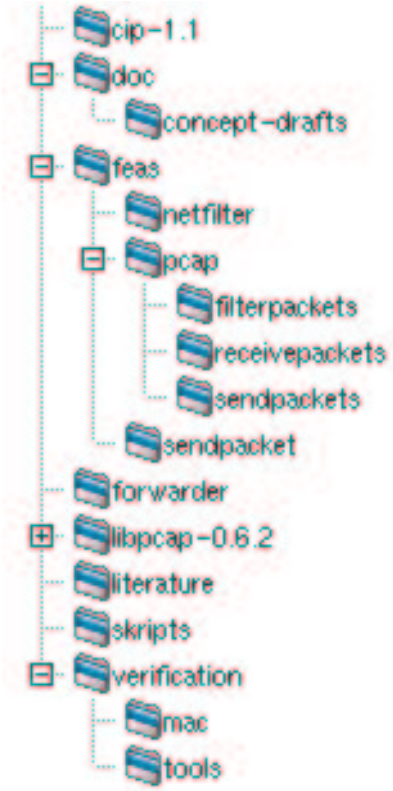
16.3 Dank

Wir bedanken uns bei der Firma FutureLAB AG, insbesondere bei Herrn Matthias Aebi und Herrn Tobias Boesch für die gute Zusammenarbeit. Ein grosser Dank gilt auch unserem Dozenten Herrn Dr. Andreas Steffen, der diese Projektarbeit mit anregenden Ideen bis zum Schluss interessant gestaltet hat.

Teil VII

Anhang

CD-ROM Verzeichnis



Glossar

AAA	Authentisierung, Autorisation & Accounting Unter Authentisierung wird das Anmelden/Abmelden eines Benutzers am “Cellular IP”-Netz verstanden, während die Autorisation der Frage nachgeht, ob ein Benutzer das Netz noch benutzen darf. Dieser Aspekt spielt schliesslich mit dem Accounting zusammen, welches Abrechnungsmodelle implementiert und beispielsweise bei Feststellung einer Kreditunwürdigkeit die Autorisation entzieht.
Cellular IP nG	Enthält und erweitert “Cellular IP” (für Details siehe Konzept)
CIP	“Cellular IP v1.1 for Linux” von der Columbia University
PCAP	Bibliothek für “Packet Capturing”
PDA	“Personal Digital Assistant”; Palm, Windows CE

Literaturverzeichnis

[1] HEROLD, Helmut: *Linux-Unix Systemprogrammierung*. 2. überarbeitete Auflage. Addison Wesley, 1999. - (ISBN 3-8273-1512-3)

[2] BECK, Michael u.a.: *Linux Kernelprogrammierung*. 6. Auflage. Addison Wesley, 2001. - (ISBN 3-8273-1659-6)

[3] *HMAC: Keyed-Hashing for Message Authentication*
IETF RFC 2104

[4] *IP Mobility Support*
IETF RFC 2002

[5] *Mobile IP Authentication, Authorization and Accounting Requirements*
IETF RFC 2977

[6] *Unreliable Guide to Hacking The Linux Kernel*
http://antarctica.penguincomputing.com/%7Enetfilter/unreliable-guides/kernel-hacking/lk-hacking-guide.html

[7] *Kernel Locking Guide*
http://antarctica.penguincomputing.com/%7Enetfilter/unreliable-guides/kernel-locking/lklockingguide.html

[8] MOUW, Erik: *Linux Kernel Procfs Guide*
http://kernelnewbies.org/documents/kdoc/procfs-guide/lkprocfsguide.html

[9] *Netfilter Hacking HOWTO*
http://antarctica.penguincomputing.com/%7Enetfilter/unreliable-guides/netfilter-hacking-HOWTO/index.html

[10] *Linux 2.4 Packet Filtering HOWTO*
http://antarctica.penguincomputing.com/%7Enetfilter/unreliable-guides/packet-filtering-HOWTO/index.html

[11] *Linux 2.4 NAT HOWTO*
http://antarctica.penguincomputing.com/%7Enetfilter/unreliable-guides/NAT-HOWTO/index.html

[12] STEINER Marc, WENDEL Thomas: *Wireless LAN basierend auf Cellular IP*. ZHW Projektarbeit.
http://www.strongsec.com/zhw/PA/PA1_Sna06_2001.pdf

[13] *Cellular IP Home Page*
http://www.comet.columbia.edu/cellularip/

[14] *Columbia IP Micro-Mobility Suite*
http://comet.ctr.columbia.edu/micromobility/

[15] VALKO, A. G.: *Cellular IP - A New Approach to Internet Host Mobility*
http://www.comet.columbia.edu/cellularip/pub/ccr99.pdf

[16] CAMPELL A. T., GOMEZ J., KIM S., TURANYI Z., WAN C-Y., VALKO A. G.: *Design, Implementation and Evaluation of Cellular IP*
http://www.comet.columbia.edu/cellularip/pub/pcs2000.pdf

[17] *IETF Internet Draft "Cellular IP"*
http://www.comet.columbia.edu/cellularip/pub/draft-ietf-mobileip-cellularip-00.txt

[18] *Linux WaveLAN IEEE 802.11 Treiber*
http://www.fasta.fh-dortmund.de/users/andy/wvlan/

[19] *Agere's ORiNOCO Home Page*
http://www.orinocowireless.com

[20] *Wireless LAN resources for Linux*
http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/

[21] *Linux-WLAN Project*
http://www.linux-wlan.com/linux-wlan/

[22] *GMP - GNU Multi-Precision Library*
http://swox.com/gmp/